



**NAVAL
POSTGRADUATE
SCHOOL**

MONTEREY, CALIFORNIA

THESIS

**ADAPTIVE EQUALIZATION FOR BURST SIGNALS IN
DISPERSIVE CHANNELS**

by

Margaret Riordan

December 2005

Thesis Advisor:

Second Reader:

Charles Therrien

R.C. Robertson

Approved for public release; distribution is unlimited.

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, Va 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY (Leave blank)			2. REPORT DATE December 2005	3. REPORT TYPE AND DATES COVERED Master's Thesis
4. TITLE AND SUBTITLE Adaptive Equalization for Burst Signals in Dispersive Channels			5. FUNDING NUMBERS	
6. AUTHORS Riordan, Margaret				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.				
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.			12b. DISTRIBUTION CODE	
13. ABSTRACT(<i>maximum 200 words</i>) In this thesis a method is developed to estimate and remove group delay caused by the transmission channel in the case of a burst signal. The signal of interest is modeled as burst white noise, and no information about any underlying symbol structure or rate is assumed. The regions of transition between the signal present and the signal not present are used to estimate the group delay of the transmission channel. Once estimated, the group delay can be removed using an all-pass filter with the proper phase characteristics.				
14. SUBJECT TERMS Adaptive Equalization, Dispersive Channels, Group Delay			15. NUMBER OF PAGES 77	16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**ADAPTIVE EQUALIZATION FOR BURST SIGNALS IN
DISPERSIVE CHANNELS**

Margaret Riordan
Civilian, Department of Defense
B.S., New Mexico State University, 1984

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN ENGINEERING SCIENCE
(ELECTRICAL ENGINEERING)**

from the

NAVAL POSTGRADUATE SCHOOL
December 2005

Author: Margaret Riordan

Approved by: Charles Therrien
Thesis Advisor

R C Robertson
Second Reader

Jeffrey B. Knorr
Chairman, Department of Electrical Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In this thesis a method is developed to estimate and remove group delay caused by the transmission channel in the case of a burst signal. The signal of interest is modeled as burst white noise, and no information about any underlying symbol structure or rate is assumed. The regions of transition between the signal present and the signal not present are used to estimate the group delay of the transmission channel. Once estimated, the group delay can be removed using an all-pass filter with the proper phase characteristics.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION	1
A.	THESIS OBJECTIVES AND ORGANIZATION	2
B.	DESCRIPTION OF THE PROBLEM	2
C.	DESIGN OF GROUP DELAY FILTERS	5
D.	THE OPTIMUM SOLUTION	12
E.	THE WIENER METHOD APPLIED TO THIS PROBLEM	16
II.	CHANNEL-CORRELATION METHOD	25
A.	STEP 1: FILTERING THE SIGNAL INTO INDIVIDUAL FREQUENCY CHANNELS	25
B.	STEP 2: CROSS-CORRELATING CHANNELS TO DETER- MINE RELATIVE OFFSETS	30
C.	STEP 3: CONVERTING RELATIVE OFFSETS INTO AN EQUALIZING FIR FILTER	34
III.	EVALUATION OF THE ALGORITHM	39
A.	EFFECTIVENESS MEASURES	40
B.	RESULTS OF TESTING THE CHANNEL - CORRELATION EQUALIZATION ALGORITHM	45
1.	No Additive Noise	46
2.	Additive Noise resulting in 13dB SNR	49
3.	Additive Noise resulting in 10dB SNR	51
IV.	CONCLUSIONS AND FUTURE WORK	55
A.	POSSIBLE ALGORITHMIC IMPROVEMENTS	55
B.	MORE TESTING	57
C.	IMPROVED MEASURES OF EFFECTIVENESS	58
	LIST OF REFERENCES	59

LIST OF FIGURES

1.	Example of a Transmitted and Received Burst Signal	4
2.	Example of a Filter to Apply a Specified Group Delay (N=15)	10
3.	Example of a Better Filter to Apply a Specified Group Delay (N=29) .	11
4.	Best Equalizing Filter for Channel Shown in Figure 2	15
5.	Wiener Filtering Applied to the Problem	17
6.	Distorting Channel Filter	20
7.	Equalizing Filter Recovered by Wiener Method	22
8.	Time Waveforms of Transmitted, Received, and Equalized Signals in the Wiener Example	23
9.	Spectrograms of Transmitted, Received, and Equalized Signals in the Wiener Example	24
10.	Result of Separating the Received Signal into Individual Frequency Channels	27
11.	Time Waveforms of Transmitted, Received, and Equalized Signals in the Channel Correlation Example	28
12.	Spectrograms of Transmitted, Received, and Equalized Signals in the Channel Correlation Example	29
13.	Result of Fitting Exponential Curves to the Upper and Lower Halves of the Frequency Range	35
14.	Equalizing Filter Recovered by Channel-Correlation Method	37

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

I.	Example of the Design of a Group Delay Filter (N=15)	7
II.	Example of the Results of Correlating Channels Against the Next Higher Chosen Channel	32
III.	Example of Accumulating Offsets to Estimate Delays	34
IV.	Measures of Effectiveness of the Equalizing Filter Shown in Figure 14 .	43
V.	Effectiveness of an Equalizing Filter Derived from Data With No Additive Noise	47
VI.	Offsets of Equalizing Filter Derived from Data with No Additive Noise	48
VII.	Effectiveness of an Equalizing Filter Derived from Data With Additive Noise $N(0,0.5)$ (SNR=13dB)	50
VIII.	Offsets of Equalizing Filter Derived from Data with Additive Noise $N(0,0.5)$ (SNR=13dB)	50
IX.	Effectiveness of an Equalizing Filter Derived from Data With Additive Noise $N(0,0.1)$ (SNR=10dB)	52
X.	Offsets of Equalizing Filter Derived from Data with Additive Noise $N(0,0.1)$ (SNR=10dB)	52

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

It is said that it takes a community to raise a child. In this case it took the efforts of several communities to bring this thesis to completion. Thanks to all who encouraged me in this project, especially those that reviewed my work and my advisor Prof. Charles Therrien. And to the first person to encourage me in this — my dad.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Transmission channels can have many different effects on the signals they carry, including delaying frequency components of the transmitted signal by different amounts, a condition known as dispersion or group delay. Adaptive equalization refers to attempts by the receiver to undo the damaging effects of a varying transmission channel based on the features of a received signal. Most adaptive equalization methods require strong statistical models of the transmitted signal including (for a digital signal) the rate at which the symbols are transmitted and information about the ‘constellation’ of possible symbols [Ref. 1, 2].

The adaptive method developed here estimates and removes the transmission group delay using the structure of the energy in the on/off transitions of the received signal. There is no need to assume a strong model for the structure of the transmitted signal; the only assumptions are that the signal arrives in bursts and that its power is spread broadly across the transmission channel. The algorithm involves filtering the received signal into narrowband frequency channels and cross-correlating these channels to determine how to align them to obtain sharp transitions. There is no attempt to compensate for any variations in the passband magnitude of the transmission channel. A detailed description of the algorithm is given in this thesis.

An integral part of developing an algorithm to remove group delay is to develop methods to *apply* specified group delay to signals: both to simulate transmission channels and to equalize for the group delay recovered by the equalization algorithm. It is also necessary to be able to quantify any improvements made by the equalizing filter. Both of these topics as well as some results of testing the equalization method are discussed in this thesis.

A. THESIS OBJECTIVES AND ORGANIZATION

The objective of this thesis is to investigate algorithms that make use of the onset and cessation of a burst signal to equalize for transmission channels that apply severe group delay. The goal is to develop an algorithm that would recover and remove inter-symbol interference (ISI) caused by non-ideal transmission channels using knowledge of the bandwidth of the signal and standard filtering and equalization techniques, and without knowledge of the symbols sent, modulation type, or baud rate.

The thesis is divided into four chapters. The first chapter gives a complete description of the problem as well as the results of some preliminary research. The second chapter gives a detailed description of the channel-correlation algorithm which shows promise of being able to equalize in the stated conditions. The third chapter describes several methods to quantify the effectiveness of an equalizing filter under the stated conditions, and presents the results of testing the channel-correlation method under differing noise conditions. Finally, the fourth chapter details conclusions drawn from this research and provides suggestions for future work.

B. DESCRIPTION OF THE PROBLEM

A dispersive transmission channel can be modeled as a Finite Impulse Response (FIR) filter. In this model the received signal $r[n]$ is the result of convolving the transmitted signal $x[n]$ with the impulse response of the channel, $h_c[n]$ which has length N_c . This model for the transmission channel is useful because it can simulate such common channel effects as suppression of certain frequencies, and delay of some frequencies relative to others. Such simulations are useful when developing and testing new communications algorithms.

An important problem in communications is to overcome the dispersive and other effects of the transmission channel. This process, known as equalization, involves developing an equalizing filter $h_e[n]$ that, when applied to the received data,

compensates for the undesired effects of the channel and produces a closer approximation of the original transmitted signal [Ref. 1, 2, 3].

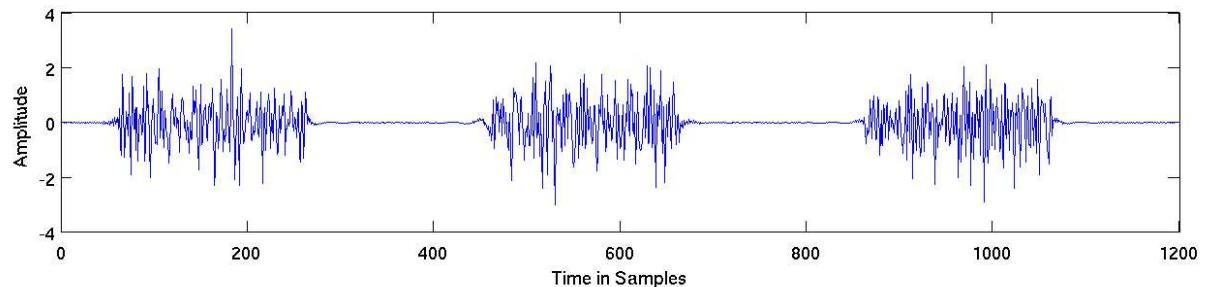
In many communication situations, the transmission channel changes over time, and to be effective an equalizing filter must change to fit each new situation. Adaptive equalization is the process of calculating the parameters of the equalizing filter based on the received signal alone. In order to do this, some type of model for the transmitted signal must be assumed. The equalizing filter is designed to modify the received signal so that it more closely fits the *a priori* model of the transmitted signal.

Often these models for adaptive equalization require information about the rate at which digital symbols are transmitted and some statistical model for the constellation of possible symbols. It is not always possible, however, to make a detailed model of the transmitted signal. It would be useful in some instances to be able to equalize a received signal without requiring knowledge of the symbol rate and constellation.

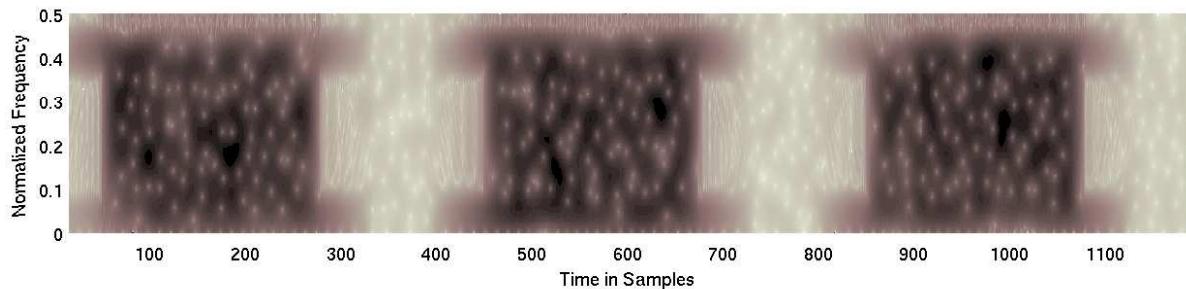
The situation investigated here is one where a detailed model of the internal signal structure is not available. All that is known is that the energy of the signal is “noise-like” in that it is spread broadly across the transmission channel and that the signal is a burst signal, turning on and off many times within the time span of the analysis. These on/off transitions are sharp in time and frequency in that all the frequency components of the signal come up at the same time and go down at the same time. A time waveform and spectrogram of such a signal is shown in Figure 1 (a) and (b). Such short burst signals are common in packetized computer-to-computer communications [Ref. 4, 5].

For this research, it is also assumed that the channel parameters vary slowly. In particular, the transmission channel is assumed to be constant for the span of data under analysis so that all the available data can be used for equalization.

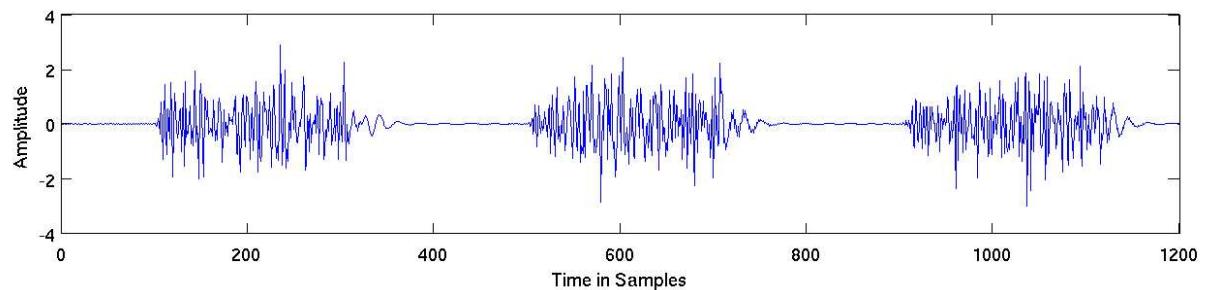
Although the signal model is not very restrictive, it is sufficient to equalize for



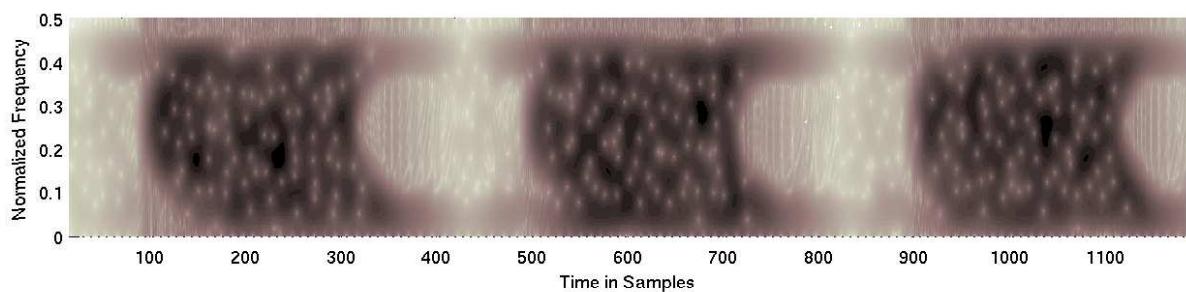
(a) Time Waveform Plot of Transmitted Signal



(b) Spectrogram of Transmitted Signal (FFT size=256 Window size=30 Overlap=29)



(c) Time Waveform Plot of Received Signal



(d) Spectrogram of Received Signal (FFT size=256 Window size=30 Overlap=29)

Figure 1. Example of a Transmitted and Received Burst Signal

severe group delay such as might be seen in poorly conditioned telephone channels [Ref. 1]. This transmission channel group delay causes the on/off transitions of the transmitted signal to be modified from straight vertical edges in the time-frequency plane to curved edges (see Figure 1 (c) and (d)). This effect is due to the transmission channel having delayed the onset of some of the signal frequency components relative to others, so that the received signal does not have a sharp transition across all frequencies at once from off to on. A similar effect happens at the time the signal turns off. The presence of transition *regions* instead of sharp edges, and the structure within these regions contains information that can be used to estimate and remove group delay.

C. DESIGN OF GROUP DELAY FILTERS

The first step in developing an algorithm to estimate and remove group delay is to create data with known group-delay characteristics. This requires designing filters with specified group delays and minimal passband shaping. The frequency-sampling design method [Ref. 6] can be used to create such filters. The basic idea is to specify the desired frequency response at equally-spaced intervals in the frequency domain, and then compute the inverse DFT of this sequence to get the impulse response of the filter. This technique is not optimal in terms of getting the most magnitude or phase change for a given length of filter, but it is straightforward to implement and allows for independent control of the magnitude and phase of the filter.

In order to describe this method in more detail, the general idea of “delay” must be refined. For this thesis, all FIR filters will have odd length, and “zero-delay” with respect to a filter will mean that the output signal starts at the central tap of the filter. As an example, if a filter has impulse response length $N = 31$ and the filter does not alter the input signal, applying 0-delay and all-pass magnitude shaping to all frequencies, then the impulse response of this filter would have taps that are all

0's except for a single 1 at position 16 (counting from 1¹). This filter would simply move a sample occurring in the input at position n to the output at position $n + 15$. This definition of 0-delay meaning to delay by $(\frac{N+1}{2} - 1)$ samples (where N is the length of the filter's impulse response and is odd) allows for filter designs that can apply both positive and negative delays to frequency components.

Since the frequency-sample points where the value of the filter's frequency response will be specified are equally spaced starting at 0 radians, the number of frequency sample points N determines which frequencies will be assigned specified values. Using normalized frequencies ranging from 0 to 2π , the frequencies which will have specified values are $\frac{2\pi}{N}(k-1)$ for $1 \leq k \leq N$. Table I shows the frequency sample locations between 0 and π that will have specified values when $N = 15$.

Using this definition of frequency delays and the fact that the frequency-sample points are evenly spaced, a MATLAB function was designed to take user input as to the number of frequency sample points, which ends of the spectrum (upper or lower frequencies) to delay, and the parameters of polynomials (degree, x and y scaling, etc.) that define the desired shape of the delay curve; and based on these inputs, to write out an array of delays $d[k]$ to be applied at the frequency-sample points. These delays are measured in (time) samples, and are floating-point numbers. For example, if a 15-point filter with linearly increasing delays at the lower half of the frequency band is specified, the output delays might be (depending on scaling parameters) as shown in Table I in the row labeled "Requested Delay."

The delays listed in Table I are in frequency order from 0 to π , so the first number (4) means that the frequency component centered around 0 radians/sec (width of $\frac{2\pi}{N}$ radians) is to be delayed 4 samples relative to the central tap of the filter. The research documented here only concentrated on real-valued signals, and all filters were designed to have real impulse responses. Therefore, all filter frequency specifications

¹Please note that, when discussing MATLAB implementations of algorithms, all indexing of filter responses and frequency bins will be 1-based.

k	1	2	3	4	5	6	7	8
Sample Frequency	$2\pi \frac{0}{15}$	$2\pi \frac{1}{15}$	$2\pi \frac{2}{15}$	$2\pi \frac{3}{15}$	$2\pi \frac{4}{15}$	$2\pi \frac{5}{15}$	$2\pi \frac{6}{15}$	$2\pi \frac{7}{15}$
Requested Delay	4.0	3.0	2.0	1.0	0.0	0.0	0.0	0.0
$ H[k] $	0.148	0.594	0.949	1.004	1.005	1.000	1.006	1.000
$\angle H[k]/\pi$	-0.667	-0.667	-0.667	-0.667	-0.800	0.933	0.533	0

Table I. Example of the Design of a Group Delay Filter (N=15)

are conjugate symmetric about the normalized frequency π ; i.e. if $H[k]$ is the specified frequency response then

$$H[(N + 2) - k] = H^*[k]$$

for $1 \leq k \leq N$.

Once the delays are specified, they can be used to create $H[k]$, the specified frequency-sampled response of the filter. The MATLAB function written for this purpose takes as input an array of delays and information about passband magnitude shape, and from these develops $H[k]$. The two parts of $H[k]$, the phase specifications $\angle H[k]$ and the magnitude specification $|H[k]|$, are calculated separately and then combined.

To calculate $\angle H[k]$, the specified delays such as in Table I must be translated into phase specifications for the filter. [Ref. 7] defines $\tau_g(\omega)$, the group delay of a filter, as the negative derivative of the phase $\Theta(\omega)$ with respect to frequency :

$$\tau_g(\omega) = - \frac{d\Theta(\omega)}{d\omega}$$

Since for this research, all work is done in discrete time and frequency, the discrete group delay for frequency component k , $\tau_g[k]$, is a function of the first difference of the phase of the filter frequency response. Therefore, to compute $\angle H[k]$, the specified delays $d[k]$ must be accumulated to give the desired phase for a particular frequency

component and normalized to convert from samples to radians:

$$\angle H[k] = -\frac{i2\pi}{N} \sum_{l=1}^{k-1} d[l] \quad (\text{I.1})$$

for $2 \leq k \leq \frac{(N+1)}{2}$ and where N , the number of frequency sample points, is equal to the filter order (number of taps). In order for the the filter impulse response to be real, the frequency component centered at 0 radians must be delayed by an integer number of samples and the phase of the frequency response of the frequencies between π and 2π must have the proper symmetry with the phases for 0 to π . Therefore, in the MATLAB function $\angle H[1]$ is set to 0 and

$$\angle H[k] = (-1)\angle H[(N + 2) - k]$$

for $\frac{(N+1)}{2} + 1 \leq k \leq N$.

The second part of computing the desired frequency-sampled response of the filter $H[k]$ is to calculate the desired magnitude shape $|H[k]|$. The MATLAB function that develops the filter frequency specifications allows for 3 different frequency bandpass magnitude shapes: all-pass, low-pass, and band-pass. The all-pass option has a flat, unit magnitude frequency response. The low-pass and band-pass options mimic standard telephone shapes. The low-pass permits frequency components from 0 to 0.825π (0Hz to 3300Hz at an 8000 samples/second sampling rate). The band-pass option permits frequency components from 0.075π to 0.825π (300Hz to 3300Hz at an 8000 samples/second sampling rate). The specific magnitude values used in the frequency specifications are calculated using MATLAB's *fir1* function and the N frequency sample points. Only the magnitude of the frequency response calculated by *fir1* is used.

Once the phase and magnitude specifications are calculated, they can be combined to form the value of the frequency response at the frequency-sampled locations:

$$H[k] = |H[k]|e^{i\angle H[k]}. \quad (\text{I.2})$$

For example, if the delays listed in Table I in the row labeled “Requested Delay” are used as input to the function that develops the frequency specifications, and a band-pass magnitude shape is specified, the magnitude and phase of the frequency specifications returned by the function would be as shown in Table I in the rows labeled $|H[k]|$ and $\angle H[k]/\pi$.

The final step is to convert the frequency-sampled specifications into an impulse response $h[n]$. This is done by taking the Inverse Discrete Fourier Transform (IDFT) of the frequency specifications:

$$h[n] = \frac{1}{N} \sum_{k=1}^N H[k] e^{i \frac{2\pi}{N} (n-1)(k-1)}$$

for $1 \leq n \leq N$. This IDFT is calculated for the exact length of the frequency specifications, so $h[n]$, the impulse response of the filter, is the same length as the number of frequency samples used.

Continuing the example shown in Table I, the impulse response resulting from taking the IDFT of the given frequency specifications is shown in Figure 2 (a). The frequency response, as specified and realized for the magnitude and phase, is plotted in Figure 2 (b) and (c), and the plot of the zeroes of this filter is shown in Figure 2 (d).

The length of the filter limits the size of the delays realizable. Given that “0-delay” for an N -long filter is interpreted to mean to delay a frequency component by $\frac{(N+1)}{2}$ samples, the span of theoretically possible delays is from $(-1)(\frac{(N+1)}{2} - 1)$ to $\frac{(N+1)}{2} - 1$. Any delay outside that range will be wrapped modulo N when $\angle H[k]$ from Equation I.1 is substituted into Equation I.2 on the previous page. In this process adding multiples of N to $d[l]$ in Equation I.1 translates into adding multiples of 2π to $\angle H[k]$ which has no effect on $H[k]$ in Equation I.2.

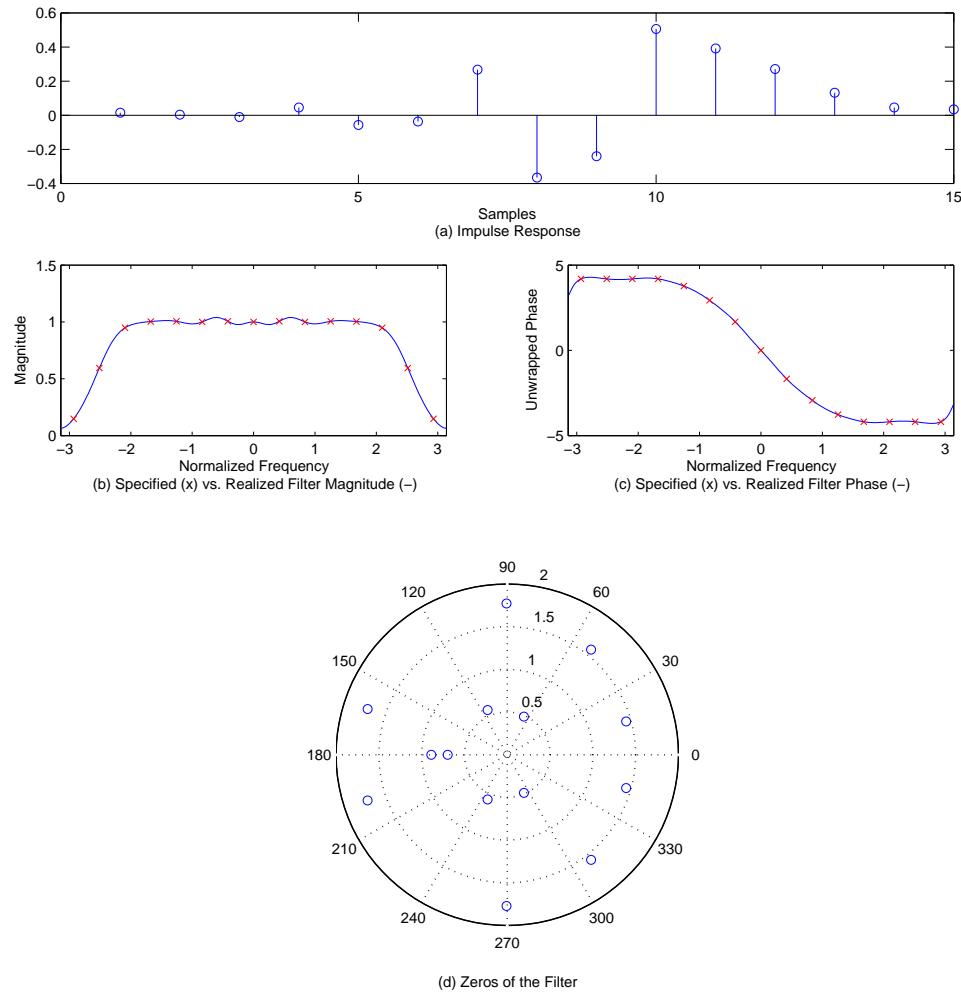


Figure 2. Example of a Filter to Apply a Specified Group Delay (N=15)

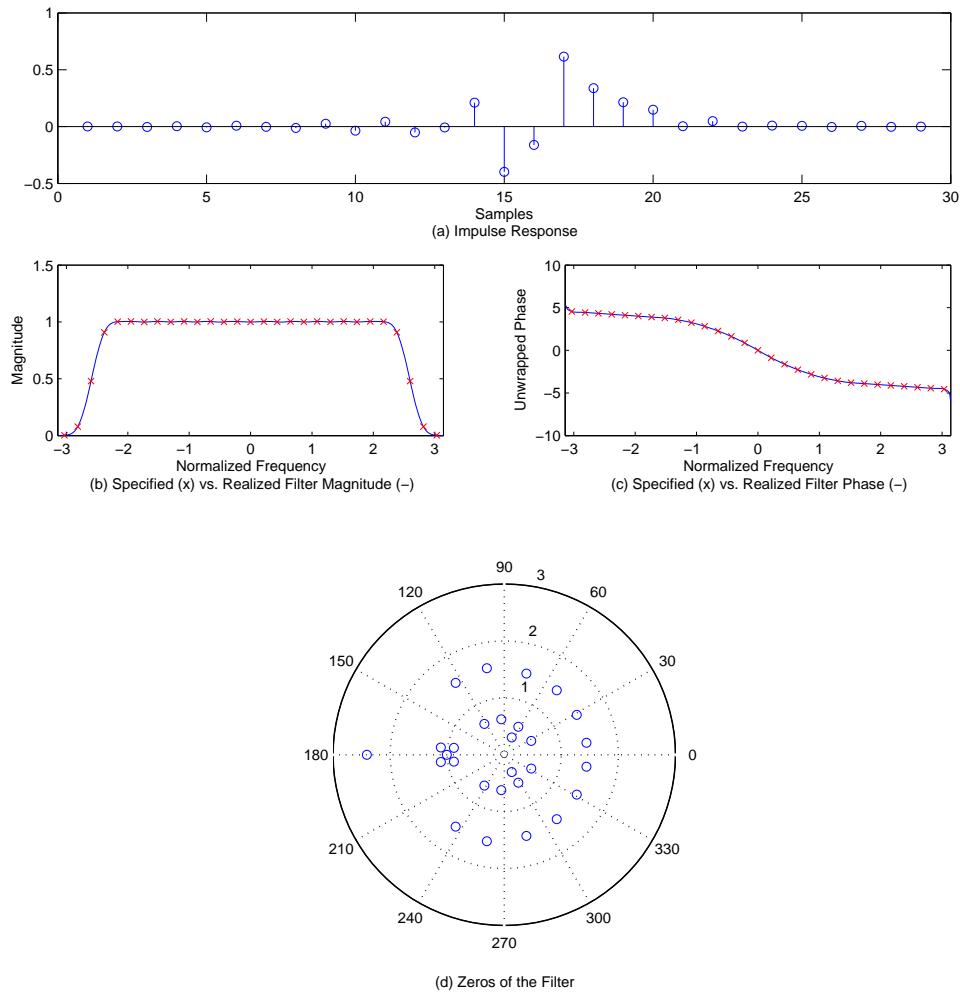


Figure 3. Example of a Better Filter to Apply a Specified Group Delay (N=29)

Implementing the filter as a *Finite Impulse Response* filter further limits the range of delays possible before the filter magnitude shape starts becoming unstable. As stated in [Ref. 6: p. 106],

“The smoother the frequency response being approximated, the smaller the error of interpolation between the sampled points.”

That is, the created filter will have a frequency response that matches the given specifications precisely at the sampling frequencies, but is only guaranteed to be a smooth function between those points. If a severe phase shift is requested, there may be some wild swings in magnitude or phase between the frequency sample points. Close examination of 2 (d) reveals that the magnitude of the filter is not perfectly flat in the passband. This ‘ripple’ can be reduced by using more frequency sample points. This will allow for finer specification of the frequency magnitude and phase, and result in a longer impulse response.

Figure 3 is a plot of the filter resulting from requesting the same delay shaping as in Table I, but with almost twice as many frequency samples ($N=29$). In this case the requested delays would be $d[k] = 4.0, 3.5, 3.0, 2.5, \dots$. Figure 3 shows that the passband ripple is much smaller for this filter. The cost for reducing passband ripple is an increase in the filter impulse response length from 15 to 29.

D. THE OPTIMUM SOLUTION

The second step in developing an algorithm to estimate and remove group delay is to form some idea of what an equalizing solution for this situation would look like. In the case where an FIR filter is used to model the group delay, the equalizing filter is straightforward; it is the same filter with the taps reversed in time. The result of first applying the distorting channel filter and then applying the reversal of the channel filter is to delay all frequencies by the same amount, thus removing any relative group delay.

Specifically, define the distorting channel filter as

$$h_c[k] = \begin{cases} a_k \in \Re, & 1 \leq k \leq N_c; \\ 0, & \text{else} \end{cases}$$

so that the received signal is

$$r[n] = \sum_{k=1}^{N_c} x[n - (k - 1)]h_c[k].$$

Then the tap-reversed version of the channel filter that is to be used as the equalizing filter is

$$h_e[k] = \begin{cases} a_{N_c - (k - 1)}, & 1 \leq k \leq N_c; \\ 0, & \text{else} \end{cases}$$

and applying it to the received signal

$$\hat{x}[n] = \sum_{k=1}^{N_c} r[n - (k - 1)]h_e[k]$$

results in \hat{x} having had all frequencies equally delayed which means no group delay.

To see this, first recall that the frequency response of the distorting channel filter is

$$H_c(\omega) = \sum_{n=0}^{N_c-1} h_c[n + 1](e^{i\omega})^{-n},$$

and the frequency response of the equalizing filter is

$$\begin{aligned} H_e(\omega) &= \sum_{n=0}^{N_c-1} h_e[n + 1](e^{i\omega})^{-n} \\ &= \sum_{n=0}^{N_c-1} h_c[N_c - n](e^{i\omega})^{-n}. \end{aligned}$$

The frequency response of the channel filter can be related to the equalizing filter response in the following manner. Start with the definition of the equalizing filter:

$$H_e(\omega) = \sum_{n=0}^{N_c-1} h_c[N_c - n](e^{i\omega})^{-n}.$$

Then, set $m = (N_c - 1) - n$ to obtain

$$\begin{aligned}
H_e(\omega) &= \sum_{m=N_c-1}^0 h_c[m+1](e^{i\omega})^{m-(N_c-1)} \\
&= \sum_{m=0}^{N_c-1} h_c[m+1](e^{i\omega})^m (e^{i\omega})^{-(N_c-1)} \\
&= \left[\sum_{m=0}^{N_c-1} h_c[m+1](e^{i(-\omega)})^{-m} \right] (e^{i\omega})^{-(N_c-1)} \\
&= H_c(-\omega)(e^{i\omega})^{-(N_c-1)}.
\end{aligned}$$

Since the filter is real, $H_c(-\omega) = H_c^*(\omega)$, so

$$H_e(\omega) = H_c^*(\omega)(e^{i\omega})^{-(N_c-1)}. \quad (\text{I.3})$$

Applying first the distorting channel filter and then the equalizing filter results in the combined system frequency response as the product of the two individual responses: $H_{sys}(\omega) = H_e(\omega)H_c(\omega)$. From Equation I.3, the magnitude of $H_e(\omega)$ is the same as for $H_c(\omega)$ for all ω , so the system frequency response is the square of the magnitude of the channel frequency response $|H_{sys}(\omega)| = |H_c(\omega)|^2$. The system phase response is the sum of the two individual phase responses and is linear:

$$\begin{aligned}
\angle H_{sys}(\omega) &= \angle H_e(\omega) + \angle H_c(\omega) \\
&= \angle \left(H_c^*(\omega)(e^{i\omega})^{-(N_c-1)} \right) + \angle H_c(\omega) \\
&= (-1)\angle H_c(\omega) + -\omega(N_c - 1) + \angle H_c(\omega) \\
&= -\omega(N_c - 1).
\end{aligned}$$

Since group delay is defined as the negative of the derivative with respect to ω of the phase response of the system, the group delay for the combined system is $(N_c - 1)$, which is constant for all frequencies.

As an example, consider the 15-point filter from Table I. Figure 4 shows the impulse response, frequency response, and the zeroes of the z-transform of the tap-reversed form of this filter. Tap-reversing an FIR filter causes each zero, $m_k e^{i\theta_k}$, to

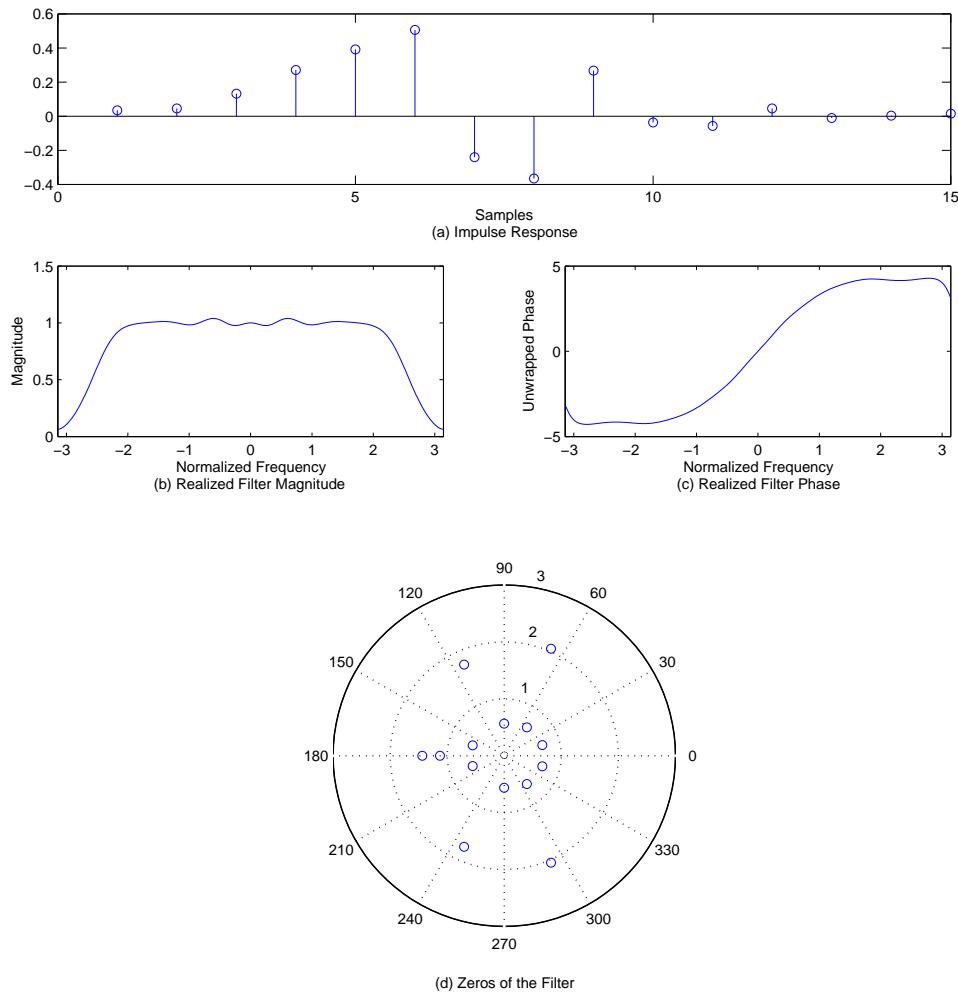


Figure 4. Best Equalizing Filter for Channel Shown in Figure 2

be moved to the conjugate reciprocal location $\frac{1}{m_k}e^{i(-\theta_k)}$. Since the filter is real, the zeroes come in pairs; i.e. for each zero $m_k e^{i\theta_k}$ there is a matching zero at $m_k e^{i(-\theta_k)}$. Thus when tap-reversing an FIR filter, what happens pictorially is each zero of the original filter inside the unit circle is moved outside the unit circle with the same angle, and each zero outside the unit circle is moved inside with the same angle. This relationship of the zeroes of the two filters can be seen in Figures 2 and 4.

The solution to group delay is to apply another group delaying filter with offsetting delays, as above. To the extent that the specific transmission channel under examination can be modeled as an FIR filter, properly designed FIR filters are the optimum solution for any group delay applied by the channel.

E. THE WIENER METHOD APPLIED TO THIS PROBLEM

A well known method for deriving an FIR channel-equalizing filter when the input signal is a random process is the Wiener method [Ref. 8]. Wiener filtering takes the received signal $x[n]$ and a model for the desired signal $d[n]$, and from these develops an FIR filter that, when applied to the received signal, produces an output $\hat{d}[n]$ that is closest to the desired signal in the mean-square sense. In the problem considered here there is not a good model for the signal, but there is a good model for the silence between signal bursts. This property can be used to formulate a Wiener filtering problem.

The application of Wiener filtering to developing an equalizing filter for bursting signals in a dispersive channel is to process the stretch of received signal between bursts where there should be no signal energy (area **b** in Figure 5). This portion of the received signal is designated as the “observation sequence” $x[n]$. The desired signal $d[n]$ would be silence. The hope is that the Wiener filter that best transforms this $x[n]$ into this $d[n]$ will also transform the non-silent area of the received signal (area **a** in Figure 5) into an approximation of its original form.

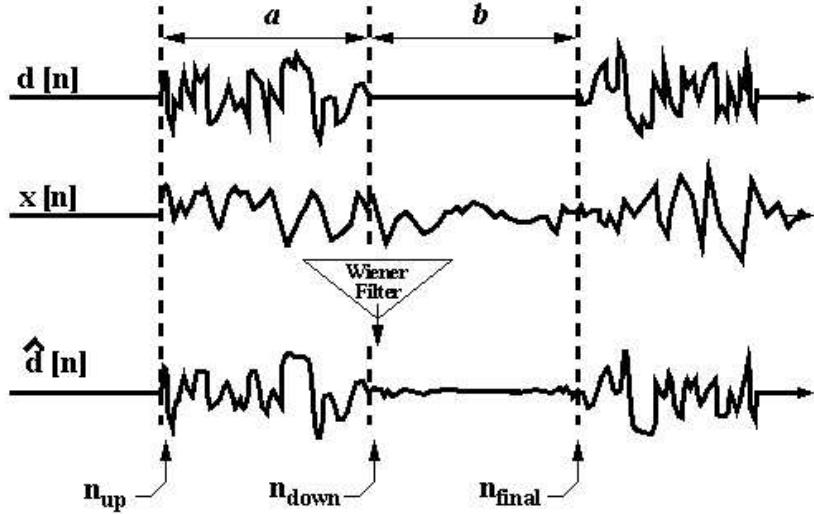


Figure 5. Wiener Filtering Applied to the Problem

In the matrix form of the Wiener-Hopf equations, the problem is formulated as finding \underline{h}_W , the N_W -long least squares solution to the over-determined set of equations

$$\left[\begin{array}{c} \underline{X} \end{array} \right] \left[\begin{array}{c} \underline{h}_w \end{array} \right] = \left[\begin{array}{c} \underline{d} \end{array} \right]$$

where \underline{X} is formed from the received data and \underline{d} is the desired result.

First, consider \underline{h}_W , the the vector of filter coefficients to be derived:

$$\underline{h}_W = \begin{bmatrix} h_W[0] \\ \vdots \\ h_W[N_W - 1] \end{bmatrix}.$$

Since an FIR filter cannot delay signal components more than its length, N_W needs to be at least as long as the longest expected delay of any frequency component in order for h_W to be able to realign the frequency components. Also, in order to have the group delay characteristics shown to be necessary in the previous section, N_W

will be non-causal and odd [Ref. 9]. If k_w is defined as $k_w = (N_W - 1)/2$, then \underline{h}_W will use k_w past and k_w future samples to derive the desired signal.

For this problem a model for the desired signal exists only for the silence area between signal bursts (area **a** in Figure 5). However, the taps of the equalizing filter should overlap into the signal burst areas on either side of the silence area in order to use all the available data. Therefore \underline{X} , the matrix of received data, has the form:

$$\underline{X} = \begin{bmatrix} x[n_d + k_w] & x[n_d + k_w - 1] & \dots & x[n_d - k_w] \\ x[n_d + k_w + 1] & x[n_d + k_w] & \dots & x[n_d - k_w + 1] \\ \vdots & \vdots & \ddots & \vdots \\ x[n_f + k_w] & x[n_f + k_w - 1] & \dots & x[n_f - k_w] \end{bmatrix}$$

where n_d is the first sample that should be 0 and n_f is the last such data sample. Notice that the data used to form \underline{X} starts k_w samples into the signal-present area (**a**), continues through the entire signal down area (**b**), and ends k_w samples into the next signal-present area.

Finally, \underline{d} is the array of desired signal values

$$\underline{d} = \begin{bmatrix} d[0] \\ \vdots \\ d[n_t] \end{bmatrix} = \begin{bmatrix} \mathbf{0} \end{bmatrix}.$$

Since the desired signal is all zeroes, the obvious solution for this system of equations is

$$\underline{h}_W = \begin{bmatrix} \mathbf{0} \end{bmatrix}.$$

This equalizing filter will exactly produce silence in the signal-down areas, but will not properly reconstruct the signal when it is supposed to be present.

To get around this problem, the equations are reformatted so that the all-zero filter is not a possible solution. This is done by setting the middle coefficient of \underline{h}_W

to 1. The set of equations to be solved becomes

$$\begin{bmatrix} \underline{X} \end{bmatrix} \begin{bmatrix} \underline{h}'_{Wa} \\ 1 \\ \underline{h}'_{Wb} \end{bmatrix} = \begin{bmatrix} \underline{0} \end{bmatrix}.$$

Then the equations can be rearranged, pulling the middle column of \underline{X} to the right-hand side of the equation to form the new received-data matrix

$$\underline{X}' = \begin{bmatrix} x[n_d + k_w] & \dots & x[n_d + 1] & x[n_d - 1] & \dots & x[n_d - k_w] \\ x[n_d + k_w + 1] & \dots & x[n_d + 2] & x[n_d] & \dots & x[n_d - k_w + 1] \\ \vdots & & & \ddots & & \vdots \\ x[n_f + k_w] & \dots & x[n_f + 1] & x[n_f - 1] & \dots & x[n_f - k_w] \end{bmatrix}$$

and the equation to be solved becomes

$$\begin{bmatrix} \underline{X}' \end{bmatrix} \begin{bmatrix} \underline{h}'_{Wa} \\ \underline{h}'_{Wb} \end{bmatrix} = - \begin{bmatrix} x[n_d] \\ x[n_d + 1] \\ \vdots \\ x[n_f] \end{bmatrix}.$$

Now, standard least-squares methods can be used to solve this set of equations and the desired equalizing filter can be formed from \underline{h}'_{Wa} and \underline{h}'_{Wb} : $\underline{h}_W = [\underline{h}'_{Wa}, 1, \underline{h}'_{Wb}]$.

This method was implemented in MATLAB and tested on simulated signals, with some success. To illustrate how this method worked, consider as the transmitted signal three bursts of noise distributed $N(0,1)$, each burst 100 samples long and separated from each other by 100 samples of silence (Figure 8 (a) and Figure 9 (a)). This signal was put through a distorting channel modeled as a 55-point FIR filter that applies a low-pass magnitude shape and a quadratically shaped delay of the lower frequencies (delays ranged from +11 samples on the upper frequency bins to -11 on the lowest frequency bin). Figure 6 shows the filter, and Figures 8 (b) and 9 (b) show the result of applying this filter to the input data.

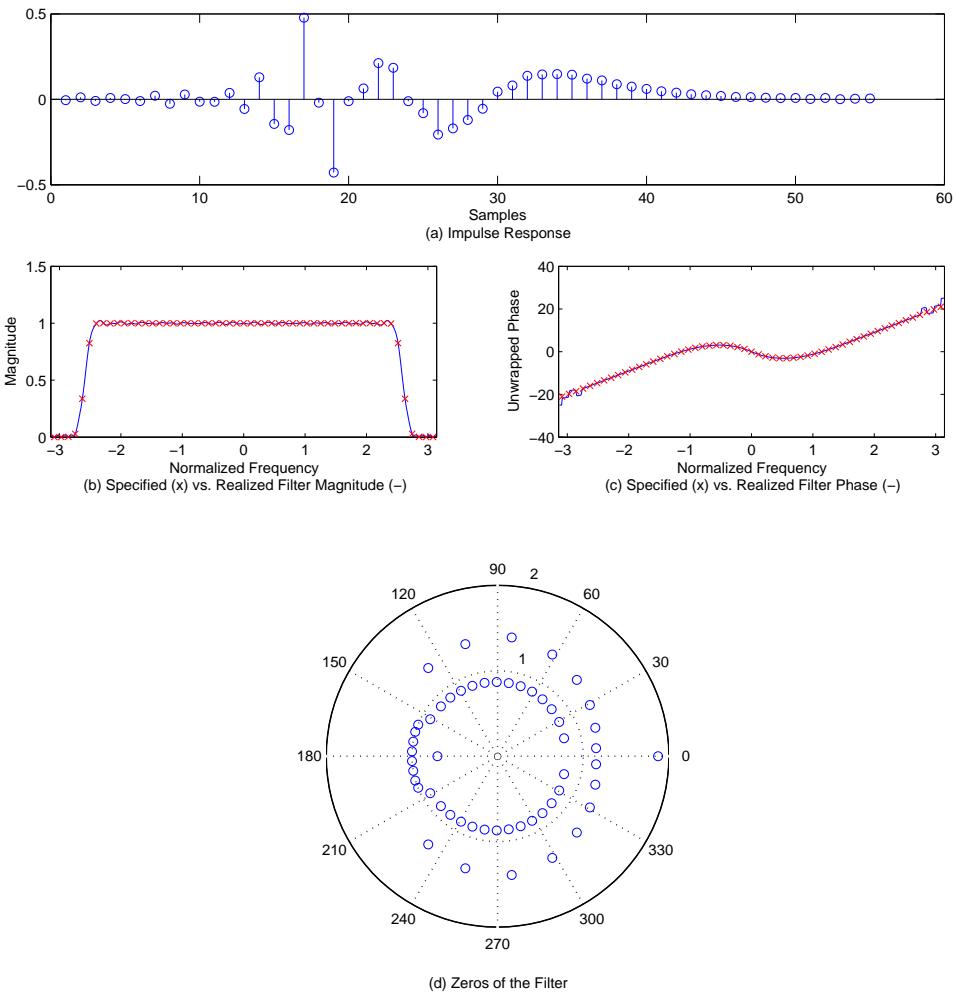


Figure 6. Distorting Channel Filter

The order of the filter to be recovered was set to 71: 16 longer than the distorting channel filter and more than three times 22, the largest frequency offset applied by the channel. All the data between the first two bursts of signal ($n_d = 101$, $n_f = 200$) were formed into a 100×71 \underline{X} array. The matrix was reformatted to force a 1 into the middle of the recovered filter and the MATLAB function *pinv* was used to find \underline{h}_W , the least-squares solution to the problem. The result is shown in Figure 7.

An examination of the Wiener-derived filter shown in Figure 7 reveals that the phase of this equalizing filter is nonlinear in the lower frequencies and approximately half of the zeroes of the filter are outside the unit circle so that this filter is a group-delaying filter and is thus of the form to properly correct for the distorting channel filter. However, this equalizing filter applies a strong emphasis in magnitude to one section of the frequency range.

Figures 8 and 9 show pictorially how this particular equalization filter affects the received data. The data in the first silence area of the received signal ((b) in both pictures) was used to derive the filter shown in Figure 7. In both figures (c) is the result of equalizing the received data with that filter. Clearly the Wiener-derived filter succeeded in driving the first silence area to zero, and to some extent the other silence areas as well. However, the magnitude shape of the equalizing filter distorted the signal by removing a great deal of the energy in the upper frequencies.

Many experiments were run varying only the seed for the random number generator used to make the input data. The magnitude of the recovered equalizing filters varied depending on what frequency components were strongest in the trailing data used to derive the filter. There was also a great deal of variance in the phase of the derived filter. Although the filter was always able to silence the stretch of data on which it was based, it had varying success on other silence sections. This is due to the random nature of the signal. The data matrix \underline{X} is based only on the samples from one silence stretch. The variations of the frequency components in individual signal bursts cause variations in the recovered equalizing filter.

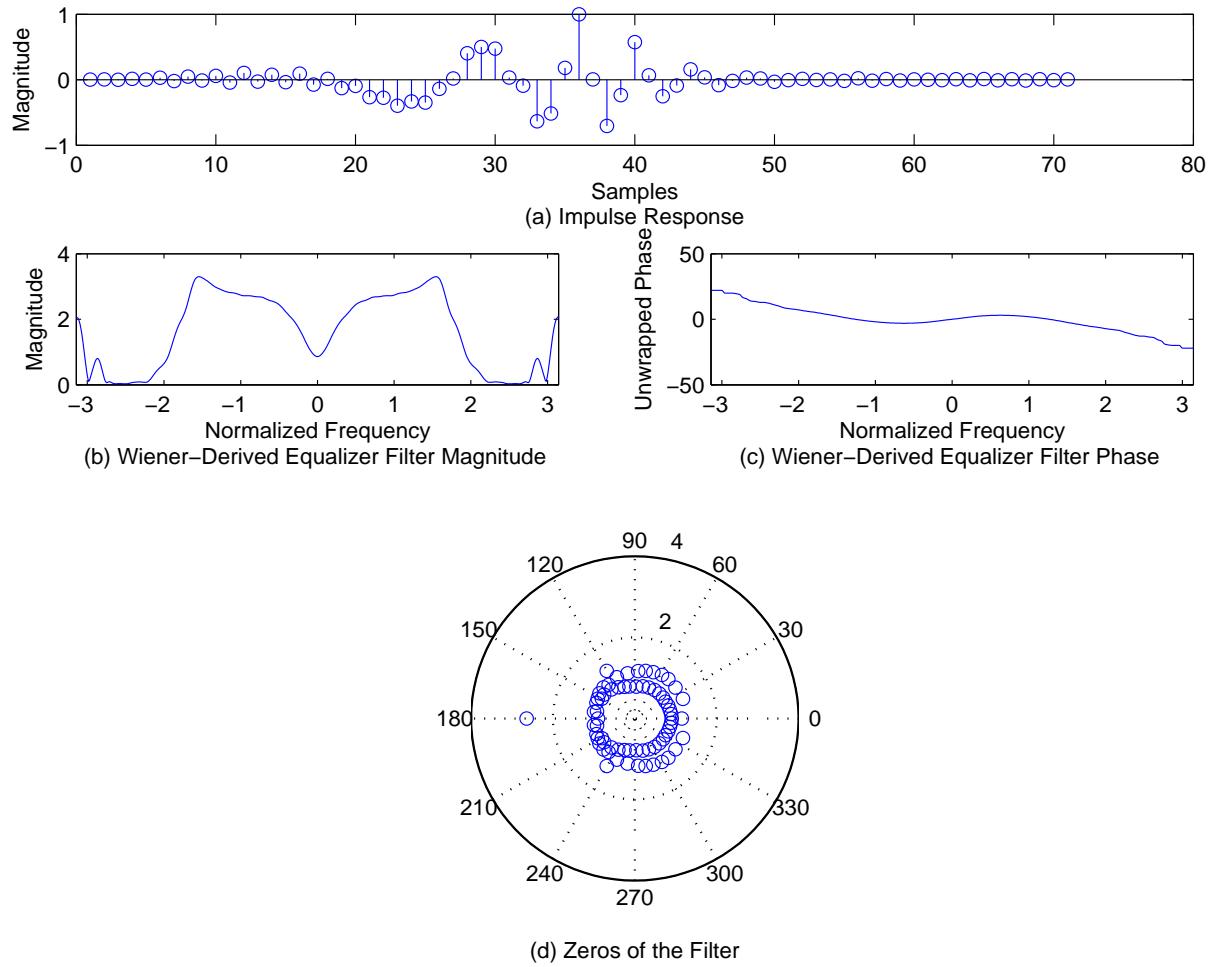


Figure 7. Equalizing Filter Recovered by Wiener Method

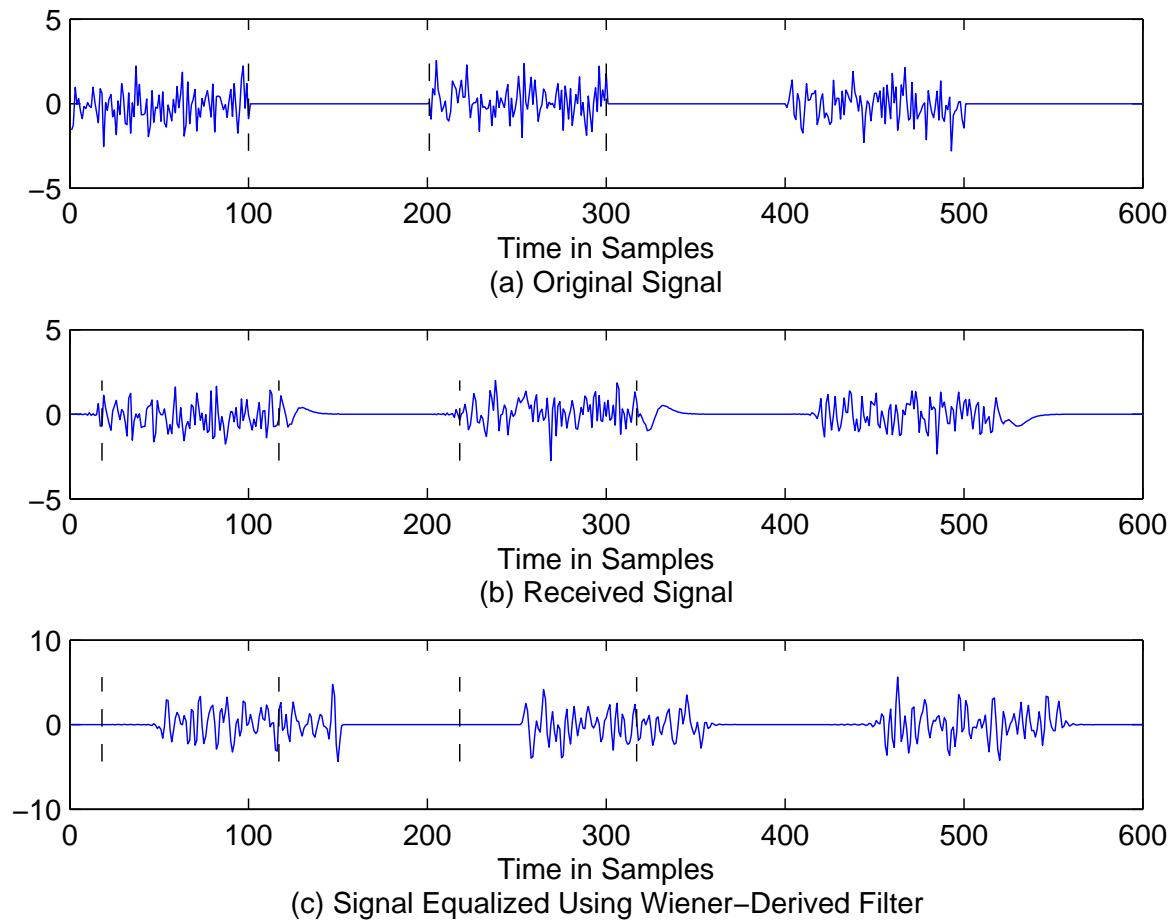
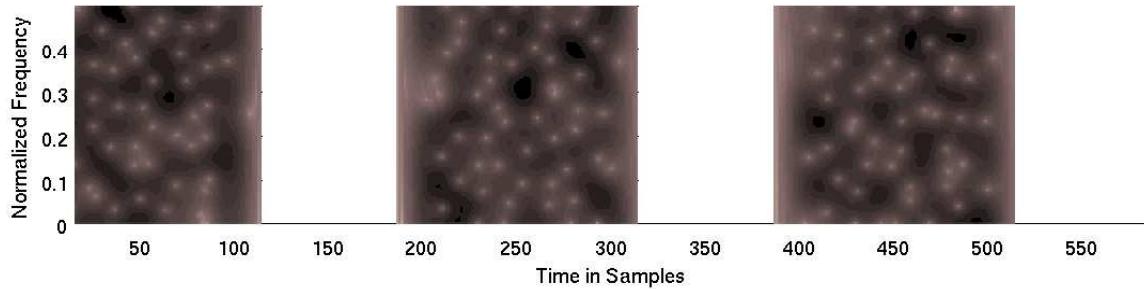
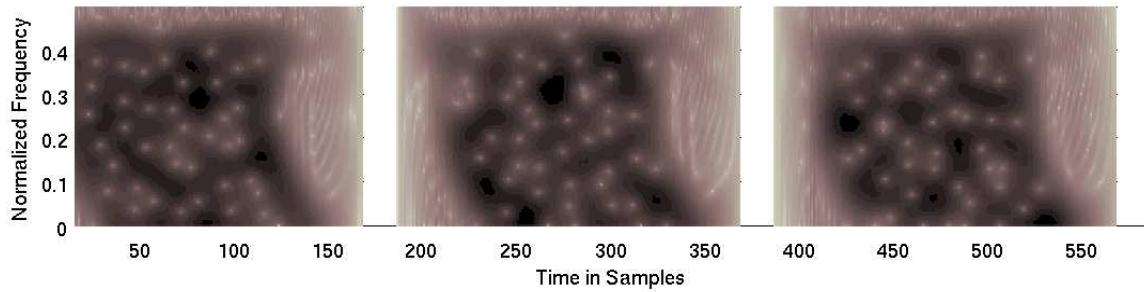


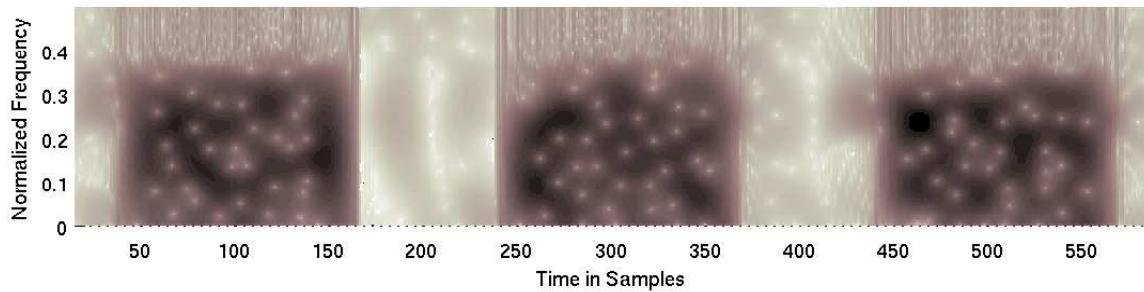
Figure 8. Time Waveforms of Transmitted, Received, and Equalized Signals in the Wiener Example



(a) Spectrogram of Original Signal (FFT size=256 Window size=30 Overlap=29)



(b) Spectrogram of Transmitted Signal (FFT size=256 Window size=30 Overlap=29)



(c) Spectrogram of Signal Equalized with Wiener-Derived Filter (FFT size=256 Window size=30 Overlap=29)

Figure 9. Spectrograms of Transmitted, Received, and Equalized Signals in the Wiener Example

II. CHANNEL-CORRELATION METHOD

The attempt to use the Wiener method to solve this problem, besides being an educational exercise in equation solving, highlighted several difficulties that should be overcome in order to have a more useful solution to the problem of burst signals in a dispersive channel. The frequency components of the signal in the on/off transition regions are variable and weak, and any method that derives an equalizing filter based on only one silence stretch will produce highly-varying results and be very sensitive to additive noise. Also, the Wiener method requires detailed knowledge of the location of two bursts in order to form the matrix of equations, and this information may not be available. When this problem was discussed with colleagues, they proposed a solution that addressed all these concerns.

The proposed method will be called the channel-correlation method. In it, the complete received signal containing multiple bursts is used so that variations in individual signal bursts and additive noise can be overcome by using more bursts. It is not necessary to know the locations of individual bursts within the received signal. The received signal is filtered into individual narrowband frequency channels, and the data in these channels are cross-correlated to find the best matching offset. The relative offsets from cross-correlation indicate how the energy in each channel has been delayed relative to the rest of the data. From this information, a group-delaying filter can be designed to bring the energy in each channel back into alignment. The details of the algorithm and its implementation in MATLAB are described in this chapter.

A. STEP 1: FILTERING THE SIGNAL INTO INDIVIDUAL FREQUENCY CHANNELS

The first step in the channel-correlation method is to filter the signal into narrowband frequency channels. This is done by computing successive Fast Fourier

Transforms (FFT's) on the time samples. The resulting streams for each channel are highly oversampled. However, the goal of this algorithm is to determine the time-alignment of the individual channels, and any procedure that would reduce or smear timing information is to be avoided. Therefore, no downsampling is done on this channelized data.

A MATLAB function was written to implement this channelization. In this function, the data is analyzed in N_A -long stretches where N_A is a power of 2. First this data is multiplied point-by-point with a window function. Then an N_A -long FFT is taken of this data, and the $\frac{N}{2} + 1$ complex numbers representing the positive frequency bins are stored as a column of an output matrix. The stretch of input data to be analyzed is then shifted by 1 sample, dropping the oldest sample and adding on the next sample from the input signal, and the analysis is repeated on this data. The input data is zero-padded so that the output channels are the same length and aligned to the input signal. The result of the function is an $(\frac{N}{2} + 1) \times T$ matrix, $C[k, n]$, where each row, $1 \leq k \leq (\frac{N}{2} + 1)$, is the complex output for a particular positive frequency bin, and each column, $1 \leq n \leq T$, aligns to a particular sample in time. N_A , the size of the analysis window, and the window function to apply to the data are user-selectable parameters.

The last part of separating the signal into frequency channels is to choose those channels that have signal-related energy so that correlations are done only using frequency channels containing sufficient signal energy. In order to simplify this process, the assumption was made that the sole source of energy in the data under analysis is from the signal of interest and that all this data has been affected similarly by the distorting channel. With this assumption, the channels with energy from the signal being equalized can be identified by first calculating the total energy in each channel, $E[k] = \sum_{n=1}^T |C[k, n]|$, and the total energy in the signal overall, $E = \sum_{k=1}^{(\frac{N}{2}+1)} E[k]$. Then channels are marked as valid for analysis starting with the most energetic and adding in channels from most to least energetic until the total

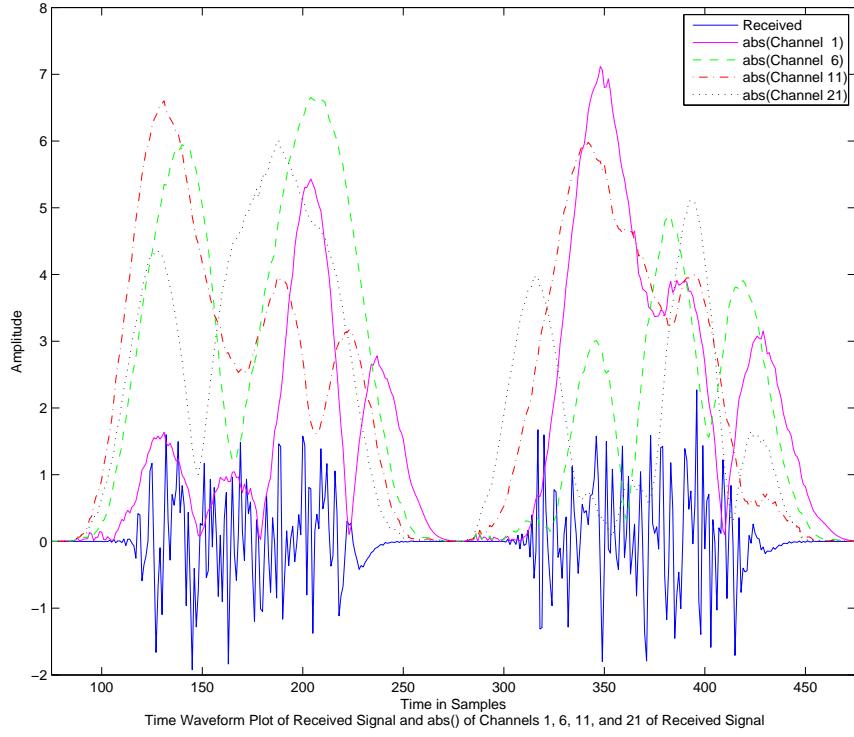


Figure 10. Result of Separating the Received Signal into Individual Frequency Channels

energy of the valid channels reaches a user-specified percentage of the total signal energy.

An example of the output of this function is shown in Figure 10. The input signal for this example is a sequence of bursts of noise distributed $N(0,1)$, each burst 100 samples long and separated from the other bursts by 100 samples of silence. The first three bursts of this data are shown in Figure 11 (a) and Figure 12 (a). This signal is then filtered by the distorting channel shown in Figure 6 and Figures 11 (b) and 12 (b) show the results.

The data shown in Figure 11 (b) was then divided into individual frequency channels using the MATLAB function described above. The analysis window used was a 64-sample-long Hamming window. The valid channels were then identified,

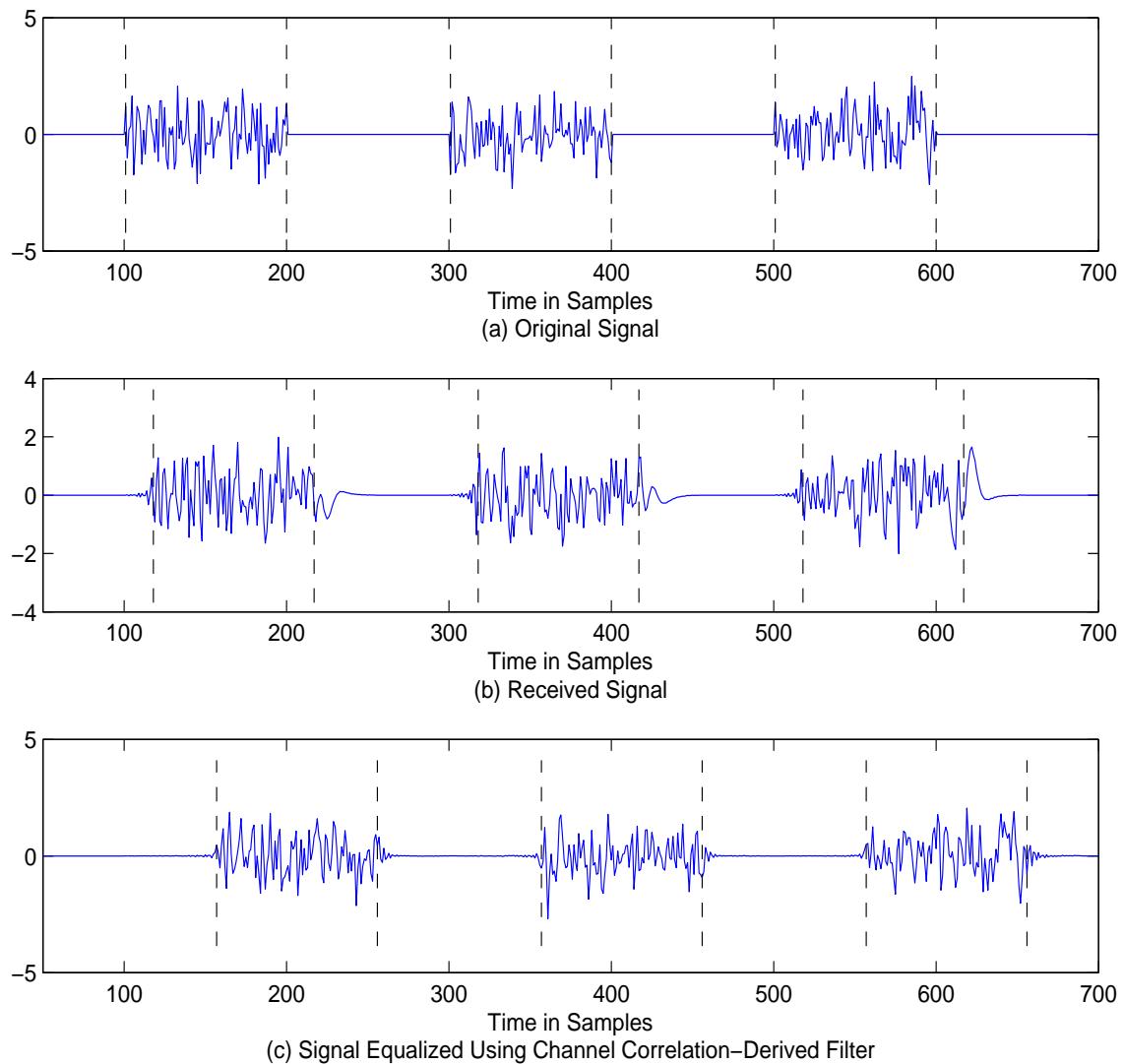


Figure 11. Time Waveforms of Transmitted, Received, and Equalized Signals in the Channel Correlation Example

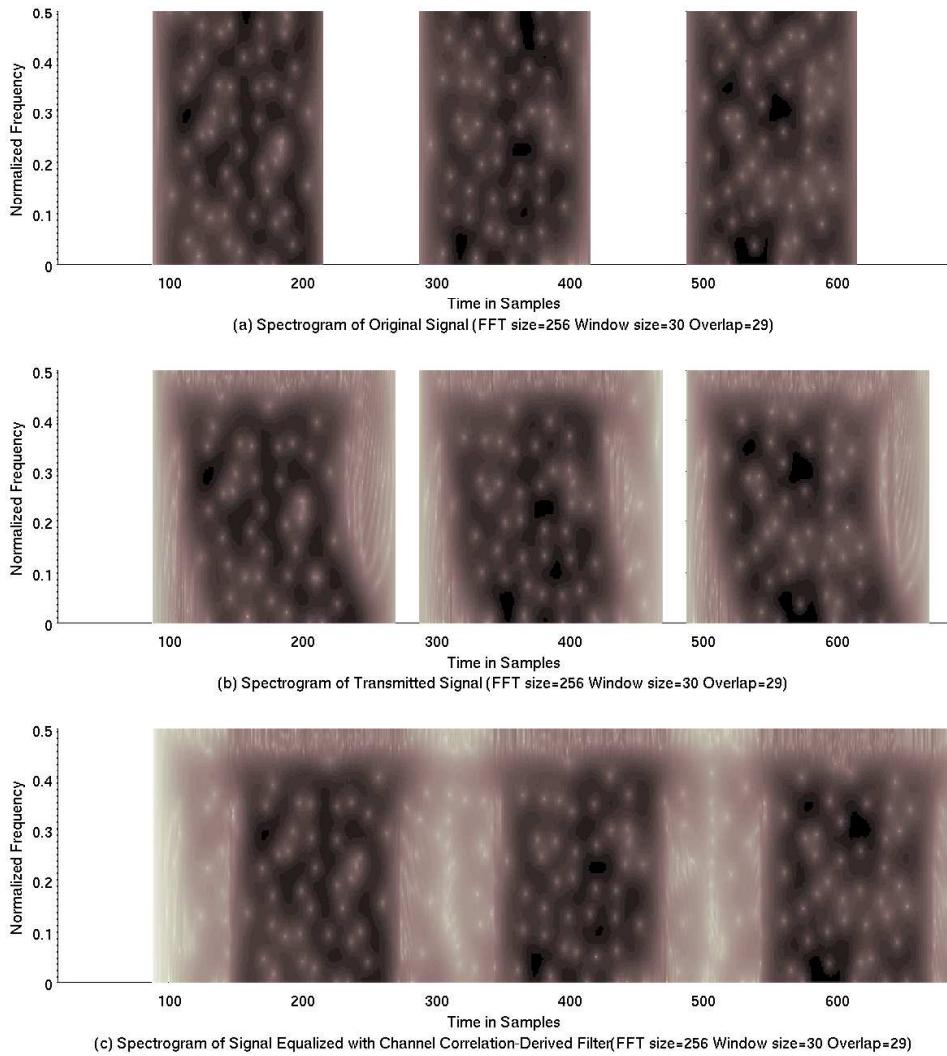


Figure 12. Spectrograms of Transmitted, Received, and Equalized Signals in the Channel Correlation Example

using 95% as the total signal energy for which to account. The result was that 26 of the 33 positive frequency channels were identified as containing valid signal data, specifically channels 1 through 26 (numbering from 1). A plot of the time waveform of the original received signal and $|C[k, n]|$ for frequency bins $k = 1, 6, 11, 21$ and samples $75 \leq n \leq 475$ is shown in Figure 10. It can be seen from the plot that the onset of the energy for the lower channels is delayed relative to the upper channels. The same delay is also seen for the cessation of the energy in each channel. The large variations in channel energy during a burst are the result of the original time samples of the signal being drawn from a Gaussian distribution.

B. STEP 2: CROSS-CORRELATING CHANNELS TO DETERMINE RELATIVE OFFSETS

Once the received signal has been separated into individual channels, the next step is to determine the alignment of the onsets and cessations of the energy in these channels. This is done by cross-correlating the magnitudes of the sequences for individual channels with those of adjacent channels; that is, cross-correlating $|C[k, n]|$ and $|C[j, n]|$ for the next valid $j > k$, summing over all n . Early experiments showed that correlating adjacent channels, $j = k + 1$, under-estimated the true offset, while correlating channels that were not adjacent improved the estimate of the transmission channel group delay. Therefore, in the MATLAB function an optional decimation may be done on the list of valid channels prior to cross-correlation. The decimation factor is user-selectable, and after decimation only the remaining channels are cross-correlated.

A MATLAB function was written to perform the channel correlations. The inputs to this function are the sequences of magnitudes for all chosen channels and a range of lags for which to compute the correlations. The range of lags to calculate is chosen large enough to detect the expected channel offsets, but not so large as to allow the data from one burst of signal to be correlated against the data from adjacent

bursts, since this might cause spurious answers to be reported. Specifying a range of lags also reduces the number of calculations that must be done.

Within this MATLAB function, the cross-correlation of each selected channel and the next selected channel higher in frequency is done using the MATLAB function *xcorr* with the default scaling option of “no scaling”. For each pair of channels correlated, the lag with the correlation that is largest in magnitude is saved along with the magnitude of the correlation. Each lag is adjusted to be an “offset” ranging positive and negative with 0 indicating that the streams correlate best without being shifted relative to each other. The sign of the offset is related to the direction of the shift necessary to align the two channels. For example, if the result from correlating channel k with channel $j > k$ is the offset $d \leq 0$, then channel j has its onsets and cessations of energy earlier in time than channel k , and channel k needs to be shifted earlier in time by $|d|$ samples relative to j in order to align with channel j . If the best-correlating offset is positive, $d \geq 0$, then channel k needs to be shifted later in time by $|d|$ samples. These measured offsets and their scores are returned by the function.

Consider the example begun in the previous section. Here the input signal consists of 500 bursts of signal with no additive noise, and the distorting channel is as shown in Figure 6. This data was channelized as described in the previous section. For the correlation calculation, the range of offsets calculated was from -40 to $+40$ samples. (If no decimation is done before the magnitudes of these sequences are correlated against each other, then each channel k with enough energy to be considered valid is cross-correlated with channel $k + 1$ as long as this channel also is a valid channel.) The strongest-correlating offsets measured in samples found by the correlation function are shown in Table II in the column labeled “Decimate by 1”. The correct offsets between adjacent channels are shown in the column labeled “Actual Offset Applied”. The measured offsets follow the general shape of the actual applied offsets, although opposite in sign because the measured offsets represent the

Analysis Channel k	Measured Offset of Channel k Relative to Next Larger Used Channel			Actual Offset Applied	
	Decimate by 3	Decimate by 2	Decimate by 1		
1	-7	-4	-1	2.62	
2			-2	2.46	
3		-5	-1	2.29	
4	-6	-4	-2	2.13	
5			-1	1.96	
6			-1	1.79	
7	-4	-4	-1	1.63	
8			-1	1.46	
9		-2	-1	1.29	
10	-3	-1	0	1.13	
11			0	0.96	
12			-1	0.80	
13	-2	-2	-1	0.63	
14			0	0.46	
15		0	0	0.30	
16	0	0	0	0.13	
17			1	0.0	
18			0	0.0	
19	-1	0	0	0.0	
20			0	0.0	
21		0	0	0.0	
22	1	0	0	0.0	
23			0	0.0	
24			0	0.0	
25	No valid channel to correlate against	0	0	0.0	
26			0	0.0	
27				0.0	
28				0.0	
29	These channels were not used			0.0	
30	used in the correlation			0.0	
31	because they did not contain			0.0	
32	enough energy			0.0	
33				0.0	

Table II. Example of the Results of Correlating Channels Against the Next Higher Chosen Channel

amount of correction needed to undo the channel-applied offsets.

If only every other channel is used in the correlation routine (the list of channels are decimated by 2), then only odd-numbered channels are used. Each valid odd-numbered channel k is cross-correlated with channel $k + 2$ if that channel is also a valid channel. The results of the correlation routine are shown in the column labeled “Decimate by 2”. If only every third channel is used, the correlation results are as shown in the “Decimate by 3” column. In the cases where a decimation greater than 1 was used, adjacent rows of the “Actual Offset Applied” column must be summed to form applied offsets comparable to the measured offsets.

Examination of Table II shows that correlating adjacent channels produces an underestimate of the actual offset. Correlating channels farther apart produces a more-accurate estimate of the offset. This is thought to be due to the use of a window on the data before calculating the FFT when channelizing the data. The window chosen (Hamming in this case) allows energy to spread across nearby FFT bins, affecting the correlation. Decimating the list of channels to be used in the correlation reduces this problem, but results in fewer points where the group delay curve is estimated. It might be possible to correlate channels 1 with 3, 2 with 4, 3 with 5 and so on, but this would produce offset estimates for overlapping frequency ranges, and the information would have to be combined properly to be useful.

Another shortcoming of this method is also apparent in Table II. Correlating the magnitude of channels against each other produces relative offsets that are measured in an integer number of samples. The actual group delay of a transmission channel does not have to have an integer number of samples of relative offset. Both this and the previously mentioned shortcomings are addressed in the next section.

Analysis Bin	1	3	5	7	9	11	13	15	17	19	21	23	25
Measured Offset	-4	-5	-4	-4	-2	-1	-2	0	0	0	0	0	0
Delay Estimate	-22	-18	-13	-9	-5	-3	-2	0	0	0	0	0	0

Table III. Example of Accumulating Offsets to Estimate Delays

C. STEP 3: CONVERTING RELATIVE OFFSETS INTO AN EQUALIZING FIR FILTER

Once the relative offsets of adjacent channels have been measured, this information can be used to develop the equalizing filter. The first step is to accumulate the relative offsets into estimates of the group delay needed to be applied at specific points in frequency. Since the channels are correlated k against $j > k$, the accumulation is done starting at the highest frequency bin and working down to the lowest. When accumulating the offsets into delays, those channels that were not allowed to correlate against other channels are considered to have zero relative offset. For example, Table III shows the result of accumulating the “Decimate by 2” column of Table II.

At this point, the group delay estimate only exists at a few points in frequency, and each estimate is an integer number of samples of delay. In order to form a more-detailed estimate of the group delay shape, a curve can be fitted to the existing data. Then the formula for the curve can be used to estimate the group delay at any point in frequency. However, fitting a curve to the data requires that a model for the expected shape of the group delay should be chosen. Based on the delay curves shown in [Ref. 1], the assumption was made that the group delay would only stay constant or increase (i.e., be monotonic) as the frequency moved away from the center of the transmission channel out to either end, thus forming a “C” or a “half-C” shape in the time-frequency plane.

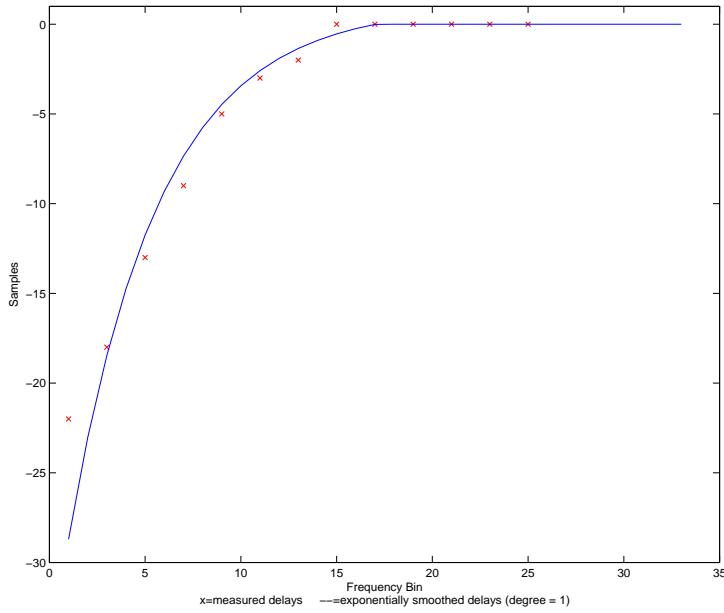


Figure 13. Result of Fitting Exponential Curves to the Upper and Lower Halves of the Frequency Range

Based on this assumption, the choice was made to fit exponential curves to the delay estimates of the upper and lower halves of the frequency range separately. In MATLAB, this is implemented by fitting a straight line to the log of the data. In order to do this, the group-delay estimates have to be shifted/rotated so that they are all positive and increasing as the frequency tends to the extremes. The amount of any shifting is stored so that any values calculated from the curves can be adjusted to fit into the original structure.

Figure 13 shows the results of taking delay estimates for bins 1-25 from Table III and fitting one exponential curve to the delay estimates for bins 1 to 17 and another to bins 19 to 25. In Figure 13 the red x's mark the delay estimates for the odd numbered bins while the blue line is the smooth curves fitted to these estimates.

Once curves are fitted to the estimated delays, they can be used to calculate the desired delay at any point in the frequency range. A length in samples can be chosen for the impulse response of the equalization filter to be applied to the data,

N_q , which should be long enough to hold the largest delay to be applied. Then the fitted curves can be used to specify the desired delay at N_q points equally spaced in frequency. This list of delays can be used as input to the programs that create FIR filters with specified group delay characteristics (described in Chapter 1, Section C). When applied to the received data, the filter created by this method will delay the frequency channels relative to each other so that the bursts of signal are more aligned in time.

Continuing the example shown in Figure 13, the exponential curves fitted to the delay estimates were used to specify the desired delays at $N_q = 65$ points equally spaced in frequency and these delay specifications were used to create the FIR filter shown in Figure 14. By design, this filter will be a group-delaying filter that will shift frequency bins in time relative to other bins.

Figures 11 and 12 show pictorially how this method performs on burst data in a dispersive channel. These figures show the first three bursts of the 500 that make up the input signal. The first line of each figure shows the original data, the second line shows the received data, and the third shows the result of applying the equalization filter shown in Figure 14. Examination of the bursts shown in these two figures reveals that the FIR filter derived by the channel correlation method produced an output signal where the onsets and cessations of signal energy are better aligned in time than the received signal. In addition the equalized waveform is more similar to the original waveform than the received waveform is.

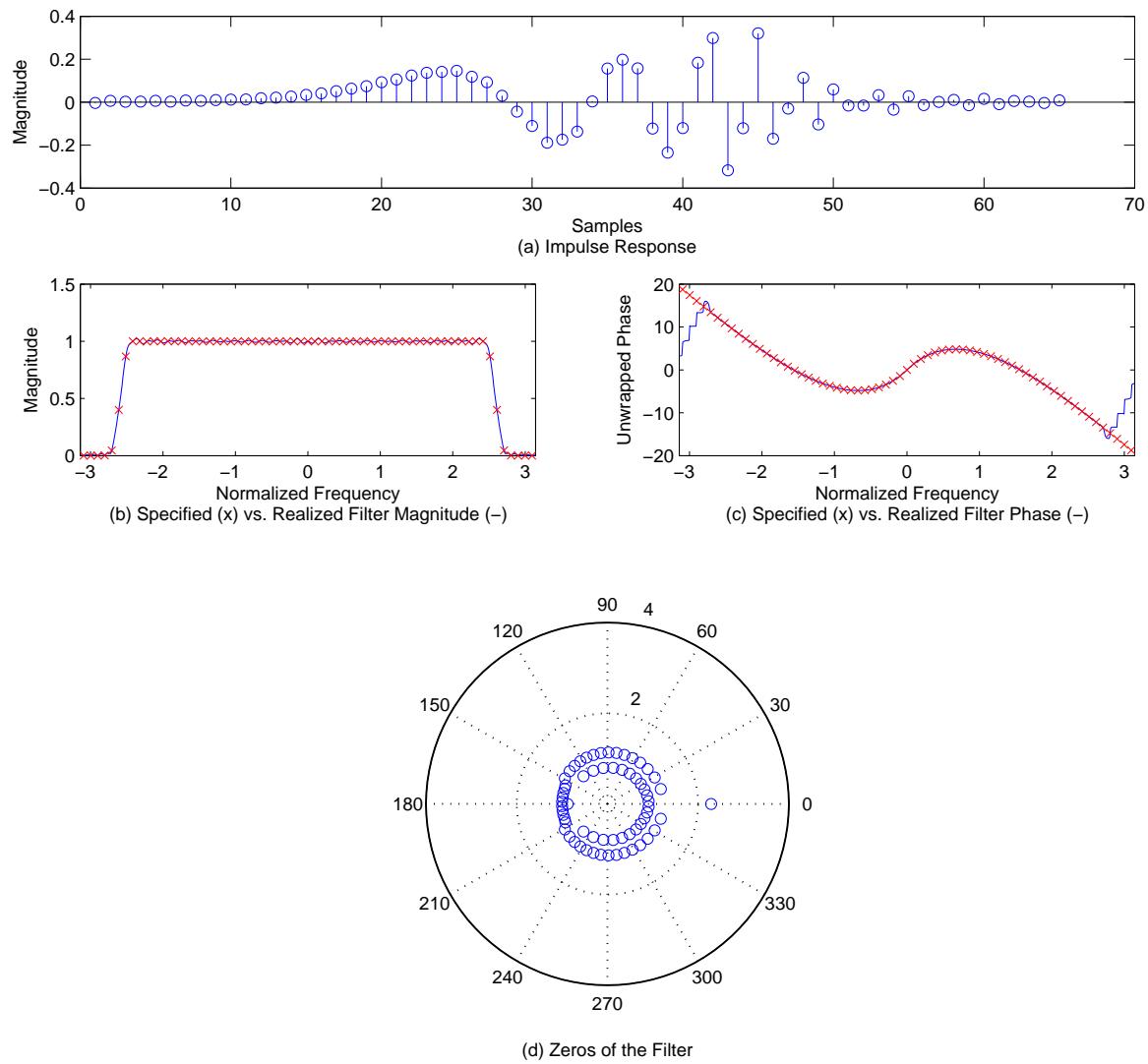


Figure 14. Equalizing Filter Recovered by Channel-Correlation Method

THIS PAGE INTENTIONALLY LEFT BLANK

III. EVALUATION OF THE ALGORITHM

Time waveforms and spectrograms comparing received and equalized signals are useful for illustrating the effects of equalization algorithms. However, it is important to be able to quantify any improvement in the data stream. In most equalization situations, the measure of improvement is the bit error rate of the demodulated signal. The goal of this research, however, was to develop an equalization algorithm that is independent of the modulation structure of the transmitted signal. Instead, the algorithm is to use the onsets and cessations of the energy of a burst signal to overcome a transmission channel with strong group delay. Therefore the effectiveness of the derived equalization filter can be measured by how well it realigns the signal energy back into bursts.

In order to quantify any improvements made by equalizing with a filter derived by the channel-correlation method, three measures of the alignment in time of the energy in a burst signal were developed. Each measure uses a different statistical feature of the data stream in an attempt to get a complete picture of how well the equalization filter reconstructs the transmitted signal. Two of the measures require additional knowledge of the structure of the original data stream, while one is based solely on the received data stream.

With these measures of improvement, the channel-correlation equalization method can be tested more thoroughly. Several tests were performed, varying the power of the added white noise. All three measures were used to gauge the effectiveness of the equalizing filter. The details of these measures of effectiveness and the results of using them to quantify the effects of the channel-correlation method are described in this chapter.

A. EFFECTIVENESS MEASURES

The first and most straightforward measure of the effectiveness of this type of equalization filter is to compare the time waveforms of the original transmitted data stream and the equalized data stream at the correct offset. The shape of the time waveform of a signal is highly dependent on the phase alignment of the frequency components that make up the signal. It is this phase alignment that is destroyed by severe group delay due to nonlinear phase of the transmission channel. The result is a received signal with a time waveform that looks very different from the original. The time waveform of an equalized signal where the frequencies had been brought back into alignment should look more like the original transmitted signal.

The method chosen to quantify the similarity between two time waveforms was the Euclidean distance between the two waveforms; that is, the sum of the squares of the differences between the samples of the waveforms. Since these data streams are delayed relative to each other, this difference should be taken at the appropriate offset, which can be found by first cross-correlating the two streams. This measure has the disadvantage that it requires precise knowledge of the time waveform of the transmitted stream, which is available only in research situations such as this. Also, it is sensitive to other transmission channel effects such as additive noise and non-flat passband magnitudes.

A MATLAB function was written to implement this method of comparison. The function takes as input parameters two data arrays, $s_1[n]$ and $s_2[n]$, and uses *xcorr* (unnormalized) to find l , the best-matching offset between the two. The function then computes the Euclidean distance between the two data arrays:

$$\text{raw_sum} = \sum (s_1[n] - s_2[n + l])^2$$

where the sum is taken over those positions where both arrays have data. In addition, the squares of those values in both arrays that were used in the calculation of the raw

sum are also summed, and used to normalize the raw sum:

$$\text{normalized_sum} = \frac{\text{raw_sum}}{\sqrt{\sum s_1[n]^2} \sqrt{\sum s_2[n]^2}}.$$

Both the raw and normalized sums are reported.

The second measure of the effectiveness of this type of equalization filter is to calculate how tightly the signal energy is concentrated into the original burst locations. Group delay spreads the onsets and cessations of a signal burst across time so that energy of the signal of interest occurs outside of the original boundaries in time of the signal burst. If the equalization filter is effective, the equalized signal will have the signal re-gathered into bursts of the same size and relative spacing as the original, although the equalized signal will be delayed by some fixed amount.

The method chosen to quantify how well the equalization filter re-gathers the signal energy into the original bursts is to compute the percentage of the total energy in the signal that is not contained in the areas where the burst signal is expected. The main disadvantage of this method is that it requires knowledge of the exact location of the signal bursts in the equalized data. This information is available in this research, since the exact structure of the original signal is known and the offset between the transmitted data and equalized data can be found by cross-correlating the two as in the previous algorithm.

A MATLAB function was written to implement this measure of effectiveness. The function takes as input a data array, $s[n]$, and the beginning and ending locations of the signal bursts. Then the percent of total energy not contained within the expected boundaries of the bursts is calculated:

$$\text{percent_energy_not_within_burst_boundaries} = 1 - \frac{\sum_i (\sum_{b_i \leq n < e_i} s[n]^2)}{\sum_{n=1}^T s[n]^2}$$

where T is the length of the data and b_i and e_i are the expected beginning and ending locations of the signal bursts. The MATLAB function reports this fraction of the total energy as a percentage.

The third measure of effectiveness was suggested by a colleague. It attempts to measure how well grouped the signal energy is into bursts without knowing the burst boundaries. This method attempts to measure how cleanly the samples divide into two groups, an energetic signal burst group, and a quiet silence group, without making hard decisions as to which group each stretch of samples belongs to. An effective equalization filter that successfully re-aligns the data into distinct signal bursts should result in samples that more cleanly separate into these two groups than the received samples do.

The method chosen to quantify how cleanly the samples divide into two groups is to estimate the entropy of the distribution of the magnitude of the sample values. In the silence areas, the magnitude of the sample values should be small and positive, while in the signal areas it should be large and positive. The more cleanly the samples break up into the two groups, the tighter the distribution of the samples will be about these two points and the lower the entropy of the distribution. The advantage of this algorithm is that it requires no extra information about the data being equalized. The disadvantage is that the amount of data that will be affected by the equalization — the transition areas — is very small compared to the entire data stream, and thus even perfect equalization will have only a small effect on the entropy of the distribution.

A MATLAB function was written to implement this measure of effectiveness. The function takes as input a data array, $s[n]$, and computes the absolute value of each sample $|s[n]|$. A histogram of these positive numbers is then formed using the MATLAB *hist* function. The number of bins into which the data is divided is the smaller of 100 or the square root of the number of data samples. The *hist* function returns the array $c_o[n]$, the number of data samples that fell into each bin n , and this number is first “flattened” by adding 1 to each bin and then normalized by dividing by the total of the counts:

$$c[n] = \frac{c_o[n] + 1}{\sum_n(c_o[n] + 1)}$$

	Received	Equalized	Improvement Received - Equalized
Distance From Transmitted Waveform			
	Raw Sum Normalized Sum	42324.8 0.94	30250.9 0.68
Energy Outside of Signal Area Total Signal Energy		5.5% 80.4%	1.3% 79.3%
			4.2%
Difference in Entropy from Transmitted Signal (transmit- ted entropy = 3.74)	0.65	0.17	0.48

Table IV. Measures of Effectiveness of the Equalizing Filter Shown in Figure 14

such that

$$1 = \sum_n c[n].$$

The flattened, normalized counts $c[n]$ are then taken as a (crude) estimate of the probability density function. The entropy of this distribution can be calculated in the standard fashion [Ref. 10]:

$$\text{entropy} = - \sum_n c[n] \log_2(c[n]).$$

The MATLAB function reports this entropy estimate.

An example of the results of using the three measures of effectiveness on a data stream equalized by the channel-correlation method is shown in Table IV. Here the original transmitted data is 500 bursts of Gaussian noise, $N(0,1)$, where each burst is 100 samples long and separated from neighboring bursts by 100 samples of silence. This data is filtered with the distorting channel shown in Figure 6 to produce the received data stream. The data is then equalized using the filter shown in Figure 14, and the measures of effectiveness are computed on the received and equalized data streams.

The output of the first effectiveness measure on this data is shown in Table IV in the row labeled ‘Distance From Transmitted Waveform’. The columns labeled

‘Received’ and ‘Equalized’ show the results of comparing the received and equalized data streams to the transmitted stream. As indicated in the ‘Improvement’ column, the equalized signal is closer in Euclidean distance to the transmitted waveform than the received is. These results indicate that the equalization filter did bring the frequencies more closely into their original phase alignment, and that the equalized time waveform is more similar to the transmitted than the received time waveform is. This agrees with the result of comparing by eye the waveforms in Figure 12. However, the equalized waveform is not a perfect match with the transmitted waveform. In this example, perfect reconstruction of the original waveform is not possible because the original signal had energy across all frequencies, while the transmission channel was band-pass in nature, cutting off any energy above 0.825π normalized frequency.

The output of the second effectiveness measure on this data is shown in the row labeled ‘Energy Outside of Signal Area’. The original data was constructed of signal bursts and perfect silence, and thus had 0% of its energy outside of signal areas. The received data has had the energy of the signal bursts spread out in time and has 5.5% of its energy outside of the signal areas. The equalized data has had its signal energy re-grouped so that only 1.3% of its energy is now outside of the signal areas. Once again, the improvement is evident.

Also listed in Table IV are the total energies of the received and equalized data as a percent of the energy of the transmitted data. Because of the low-pass magnitude shape of the channel, the received data only has 80.4% of the energy of the transmitted data. The equalizing filter also has a low-pass magnitude shape and thus the equalized data has slightly less energy than the received data. However, this change in overall energy is very small, so that the reduction in energy outside of the expected signal areas cannot only be due to removing energy. Instead, the equalization filter must have moved energy back into place.

The output of the third effectiveness measure on this data is shown in the row labeled ‘Difference in Entropy from Transmitted Signal’. The entropies of the

received and equalized data streams given on this row have been scaled by subtracting the entropy of the transmitted data (3.74 in this case) from them. Since the received and equalized data streams have greater entropy than the transmitted data, both these differences are positive. However the entropy of the equalized data is closer to that of the transmitted data, indicating that the samples of the equalized stream are better grouped into distinct signal and silence areas and showing improvement for this score as well.

B. RESULTS OF TESTING THE CHANNEL - CORRELATION EQUALIZATION ALGORITHM

With measures of effectiveness of the equalization algorithm having been developed, tests can be performed for the algorithm in the presence of additive noise. Three test sets are documented here, each set consisting of 25 individual runs. For each run within a test set the input parameters are the same:

- Each input data stream has 500 bursts of signal, each burst consisting of 100 samples of Gaussian noise, $N(0,1)$.
- Each signal burst is separated from other bursts by 100 samples of silence.
- Each data stream is filtered with the distorting channel shown in Figure 6.
- After applying the distorting channel, Gaussian noise is added, $N(0,\sigma^2)$.

This forms the received data stream. The only difference among the individual input data streams is the seed of the random number generator used to create the signal bursts and the additive noise. The only difference among the three test sets is the power of the noise added to the data, σ^2 . For the first test set no noise was added to the data. For the second test set σ^2 was set to 0.05 for a SNR of 13dB. For the third test set σ^2 was set to 0.1 for an SNR of 10dB.

The channel-correlation method is then performed on the received data to recover an equalizing filter. The analysis parameters used are as follows:

- The channelization FFT size is 64.

- A Hamming window is applied to the data before the FFT is computed.
- The total energy for which to account when determining active channels varies with the level of additive noise.
- A decimation factor of 2 is used on the channelized streams.
- The lag range of -40 to 40 samples is used when correlating the channelized streams.
- The smoothing log polynomial degree used when fitting a polynomial to the offsets is of degree 1 (a straight line is fitted to the log of the data).
- The equalizing filter constructed is 257 samples long.
- The equalizing filter is designed with a low-pass magnitude shape.

The equalizing filter is then applied to the received data to form the equalized data stream.

The three measures of improvement are then calculated on the received and equalized streams. The scores for each stream and the improvements gained by equalizing (received – equalized) are recorded for each test. After 25 runs, the average and variance for each score on each stream, and the average and variance for the improvement is reported. As an extra check on the process, the delay applied by each equalization filter is recorded for several selected points in frequency, and the average difference in delay between adjacent frequency locations is reported and compared to the true difference in delay applied by the distorting channel filter.

1. No Additive Noise

The first test was run with no additive noise ($\sigma^2 = 0$). For this test, the percent of total energy for which to account when identifying valid channels was set at 99%. The results of the effectiveness measures calculated on this data are shown in Table V. This table demonstrates that, as expected, this equalization method works well in the absence of noise, creating a filter that improves the signal quality under all measures. The most telling measure is the percentage of energy outside of the signal areas. 80% of the energy that the channel had scattered outside of the signal

	Received		Equalized		Improvement	
	Mean	Variance	Mean	Variance	Mean	Variance
Distance From Trans. Waveform Normalized Sum	0.95	0.00	0.64	0.05	0.30	0.05
Energy Outside of Signal Area	5.52%	0.03%	1.31%	0.14%	4.21%	0.18%
Difference in Entropy from Transmitted	0.66	0.00	0.50	0.01	0.16	0.01

Table V. Effectiveness of an Equalizing Filter Derived from Data With No Additive Noise

areas is moved back into place by the equalization filter, leaving only a small amount of energy in the areas that should be silence.

A final check on the equalization method in the case of no additive noise is shown in Table VI. Here each row represents one of the chosen points in frequency where the delay applied by the equalization filter was recorded for each test run. However, it is not the absolute delay applied by the equalization filter that is important, but the relative offsets between frequency components that needs to be correct in order to bring the energy of the signal into alignment. Therefore, the delay applied by the equalization filter at chosen frequency $k + 1$ was subtracted from the delay applied at frequency k to form the offsets for the selected frequency points, and the mean and variance of these were calculated across all test runs. The true offset applied by the channel is shown in the column labeled “Channel Offset”. Because the channel filter is the same, this channel offset is the same as the “Actual Offset Applied” shown in Table II, just calculated for differently spaced points in frequency.

For example, consider the the row labeled $2\pi\frac{0}{32}$ in Table VI. This row shows that the average offset in samples between chosen frequency $2\pi\frac{0}{32}$ and $2\pi\frac{1}{32}$ applied by the equalization filter is -8.95. That is, on average, the equalization filter moved the frequency component at $2\pi\frac{0}{32}$ 8.95 samples earlier in time relative to the frequency component at $2\pi\frac{1}{32}$. As shown in the final column of this table, the distorting channel

Frequency	Equalization Filter Offset Mean	Equalization Filter Offset Variance	Channel Offset
$2\pi \frac{0}{32}$	-8.95	2.65	5.08
$2\pi \frac{1}{32}$	-6.03	0.79	4.42
$2\pi \frac{2}{32}$	-4.07	0.25	3.75
$2\pi \frac{3}{32}$	-2.75	0.10	3.09
$2\pi \frac{4}{32}$	-1.86	0.05	2.42
$2\pi \frac{5}{32}$	-1.27	0.04	1.76
$2\pi \frac{6}{32}$	-0.86	0.03	1.09
$2\pi \frac{7}{32}$	-0.53	0.40	0.43
$2\pi \frac{8}{32}$	0.03	0.11	0
$2\pi \frac{9}{32}$	0.09	0.10	0
$2\pi \frac{10}{32}$	0.15	0.12	0
$2\pi \frac{11}{32}$	0.22	0.19	0
$2\pi \frac{12}{32}$	0.30	0.34	0
$2\pi \frac{13}{32}$	0.41	0.67	0
$2\pi \frac{14}{32}$	0.56	0.35	0

Table VI. Offsets of Equalizing Filter Derived from Data with No Additive Noise

filter actually moved the frequency component at $2\pi \frac{0}{32}$ 5.08 samples later in time than the frequency component at $2\pi \frac{1}{32}$; therefore, a perfect equalization filter would apply an offset of -5.08 samples at this point in frequency.

Table VI demonstrates that the channel-correlation method is recovering a filter that is closely related to the actual channel filter. The offsets of the recovered filter are approximately the same in magnitude and opposite in sign compared to the channel offsets, so the equalization filter should do a good job of removing the effects of the channel. The variance of the recovered offsets is also small, less than one tenth of the mean, except for a few cases.

The large variances for rows $2\pi \frac{0}{32}$ and $2\pi \frac{7}{32}$ are an artifact of the curve fitting. Row $2\pi \frac{0}{32}$ is at the end of the available data to which the curve must fit so the fit

of the curve at that point is not as good as at other points. Row $2\pi\frac{7}{32}$ is the offset across the center point of the frequency band and thus is the point where the lower fitted curve meets the upper fitted curve. In the curve fitting, there is no check that the two curves meet smoothly at that point, so the resulting overall filter shape may be uneven at that point.

The large variances for rows $2\pi\frac{12}{32}$, $2\pi\frac{13}{32}$, and $2\pi\frac{14}{32}$ in the table are due to the lack of energy at the higher frequencies. $2\pi\frac{12}{32}$ is the last frequency component before the channel filter cutoff of 0.825π normalized frequency. Thus, similar to row $2\pi\frac{0}{32}$, the curve fit is not as good at that point. Above 0.825π normalized frequency, there is no signal energy, so there is no information about how the channel delays are shaped, nor does it matter what the equalization filter does to that region.

2. Additive Noise resulting in 13dB SNR

The second set of tests performed used additive noise with $\sigma^2 = 0.05$, giving an SNR of 13dB. The results are shown in Tables VII and VIII. For this set of tests the percent of total energy for which to account when determining valid channels was set to 90%. All other parameters of the equalization algorithm were kept the same.

The effect of the additive noise on the measures of effectiveness can clearly be seen in Table VII. For all the measures, both the received and equalized waveforms measured worse (larger scores) than in the noise-free tests, and the variance of all the scores has increased. The added noise makes it impossible for even a perfect equalizing filter to modify the received data stream so that it matches the transmitted stream. The equalizing filter will only move the added noise around in time and, because it has a low-pass magnitude shape, remove any energy above 0.825π in frequency.

This affects the measures shown in the rows labeled ‘Distance from Trans. Waveform’ and ‘Difference in Entropy’ the most. In the case of the first measure, as the power of the added noise increases, the percentage of the score due to it increases. In the case of the entropy measure, as the power of the added noise increases, the probability density function of the samples spreads out more around the means of the

	Received		Equalized		Improvement	
	Mean	Variance	Mean	Variance	Mean	Variance
Distance From Trans. Waveform Normalized Sum	1.00	0.00	0.82	0.07	0.18	0.07
Energy Outside of Signal Area	10.41%	0.03%	6.13%	0.44%	4.27%	0.45%
Difference in Entropy from Transmitted	1.35	0.00	1.25	0.00	0.10	0.01

Table VII. Effectiveness of an Equalizing Filter Derived from Data With Additive Noise $N(0,0.5)$ (SNR=13dB)

Frequency	Equalization Filter Offset		Channel Offset
	Mean	Variance	
$2\pi \frac{0}{32}$	-8.16	4.46	5.08
$2\pi \frac{1}{32}$	-5.60	1.43	4.42
$2\pi \frac{2}{32}$	-3.86	0.49	3.75
$2\pi \frac{3}{32}$	-2.67	0.20	3.09
$2\pi \frac{4}{32}$	-1.85	0.10	2.42
$2\pi \frac{5}{32}$	-1.29	0.07	1.76
$2\pi \frac{6}{32}$	-0.90	0.04	1.09
$2\pi \frac{7}{32}$	-0.62	0.73	0.43
$2\pi \frac{8}{32}$	0.09	0.25	0
$2\pi \frac{9}{32}$	0.25	0.26	0
$2\pi \frac{10}{32}$	0.44	0.42	0
$2\pi \frac{11}{32}$	0.70	0.93	0
$2\pi \frac{12}{32}$	1.08	2.39	0
$2\pi \frac{13}{32}$	1.66	6.54	0
$2\pi \frac{14}{32}$	2.56	18.40	0

Table VIII. Offsets of Equalizing Filter Derived from Data with Additive Noise $N(0,0.5)$ (SNR=13dB)

two types of samples. In either case, since the equalizing filter only slightly effects the added noise, there is less the equalizing filter can do to improve the score.

The added noise has less effect on the second measure shown in the ‘Energy Outside of Signal Area’ row in Table VII. Here there is a larger percent of signal energy in the silence areas, both before and after equalization. However, the equalization filter affects approximately the same amount of improvement for this measure, 4%, for this data just as for the data with no additive noise. However, the variance in the improvement is greater, due to the greater variance among the recovered filters.

The effect of the added noise on the shape of the recovered filters can be seen in Table VIII. Here the means of the offsets are very similar to the means of the filters recovered from data with no additive noise as shown in Table VI. However, the variance has doubled at all points, indicating that the noise is affecting the accuracy of the channel-correlation method.

3. Additive Noise resulting in 10dB SNR

The last set of tests done used additive noise with $\sigma^2 = 0.10$, resulting in an SNR of 10dB. These results are shown in Table IX and Table X. For these tests the percent of total energy for which to account when determining valid channels was set to 90%, while all other algorithm parameters were kept the same. The results for this test indicate that the channel-correlation method is beginning to break down at this level of noise.

Once again, the additive noise affected the measures of effectiveness. Table IX shows that both the received and equalized data streams are less similar to the transmitted data than when there is no additive noise. However, the improvement provided by equalization is the same for the ‘Distance From Trans. Waveform’ and the ‘Energy Outside of Signal Area’ measures as for the SNR=13dB case. Only the entropy measure is showing a change in the amount of improvement. For this level of added noise, the entropy measure indicates no improvement provided by the equalization filter.

	Received		Equalized		Improvement	
	Mean	Variance	Mean	Variance	Mean	Variance
Distance From Trans. Waveform Normalized Sum	1.05	0.00	0.90	0.06	0.15	0.06
Energy Outside of Signal Area	14.32%	0.03%	10.01%	0.56%	4.32%	0.59%
Difference in Entropy	1.40	0.01	1.42	0.01	-0.02	0.01

Table IX. Effectiveness of an Equalizing Filter Derived from Data With Additive Noise $N(0,0.1)$ (SNR=10dB)

Frequency	Equalization Filter Offsets		Channel Offsets
	Mean	Variance	
$2\pi \frac{0}{32}$	-9.26	6.84	5.08
$2\pi \frac{1}{32}$	-6.29	2.19	4.42
$2\pi \frac{2}{32}$	-4.30	0.71	3.75
$2\pi \frac{3}{32}$	-2.95	0.26	3.09
$2\pi \frac{4}{32}$	-2.03	0.12	2.42
$2\pi \frac{5}{32}$	-1.40	0.07	1.76
$2\pi \frac{6}{32}$	-0.98	0.05	1.09
$2\pi \frac{7}{32}$	0.03	1.96	0.43
$2\pi \frac{8}{32}$	-0.06	0.39	0
$2\pi \frac{9}{32}$	0.06	0.32	0
$2\pi \frac{10}{32}$	0.17	0.35	0
$2\pi \frac{11}{32}$	0.30	0.48	0
$2\pi \frac{12}{32}$	0.45	0.79	0
$2\pi \frac{13}{32}$	0.65	1.40	0
$2\pi \frac{14}{32}$	0.90	2.62	0

Table X. Offsets of Equalizing Filter Derived from Data with Additive Noise $N(0,0.1)$ (SNR=10dB)

Table X demonstrates that the channel-correlation method is still able to provide a rough approximation to the channel delay shape despite the added noise: the overall delay shape of the equalization filter mimics that of the channel. Some of this is due to fitting exponential curves to the recovered channel offsets to smooth the data. However, even though the same curve fitting is applied to both the upper and lower halves of the frequency range, the resulting upper and lower curves have very different shapes, indicating that the the delay estimation method is responding to the channel shape and not just forcing a smooth curve onto random data.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. CONCLUSIONS AND FUTURE WORK

The channel-correlation method for equalizing burst signals in dispersive transmission channels shows promise. In the few tests documented here, the method is able to estimate transmission-channel group delay without assuming anything about the structure of the signal except that it is a burst noise-like signal. The algorithm does not require information about the locations of the bursts. The algorithm has been shown to work in the case of significant added Gaussian noise. In addition, two of the three measures of effectiveness developed here for group-delay equalizing filters correspond well with the actual performance of the filter, as measured by comparing the frequency offsets applied by the equalizing filter to the actual offsets applied by the transmission channel.

Much more research about this equalization method needs to be done, however, before it can become a robust algorithm. There are many aspects of the channel-correlation algorithm that could be improved, much more testing of the algorithm under more realistic conditions is necessary to guide the development of this algorithm, and further research into effectiveness measures is needed. In this chapter, ideas for all three areas for future research are proposed.

A. POSSIBLE ALGORITHMIC IMPROVEMENTS

The first area that needs further study is the implementation details of the algorithm. For the initial development of the channel-correlation method, many of the decisions about implementation details were made based mainly on ease of design. These choices should be carefully reexamined and alternatives tried.

The first specific implementation detail to reexamine is the method used to filter the data into separate frequency channels. In the algorithm described in this thesis, the channelization was done by applying a Hamming window to the data and then computing an FFT of this windowed data. However, experiments showed that

later steps in the algorithm were sensitive to the type of windowing function used, and this implementation detail should be more fully examined. It is also possible that a true FFT-based filter bank algorithm [Ref. 7] would be more successful at separating the energy cleanly into distinct channels, although the long band-pass filters used in FFT-based filter banks might smear out the onsets and cessations of energy that are critical to this algorithm.

Any improvement in separating the data into individual frequency channels will affect the next step of the algorithm — the cross-correlation of the channels. In the algorithm described here, the number of channels of data was reduced before correlation. This decimation was done to offset the underestimation of the relative delay that was noticed when correlating adjacent channels. However, this underestimation was probably due in part to the use of a Hamming window in the channelization. Hamming windows have good side-lobe suppression, but wide main lobes. Therefore there will be a great deal of overlap in energy between channels adjacent in frequency. It may be possible that by using a better method of channelizing the data, it won't be necessary to throw channels away.

Another algorithmic detail of the correlation of the channelized data that requires more study is deciding which channels to cross-correlate and how to combine this information. In the algorithm described here, each selected channel was cross-correlated with only two other selected channels: the next selected channel lower in frequency and the next selected channel higher in frequency. This made the estimation of the channel delay shape straightforward since for each separate frequency range there was only one estimate of the delay offset across that range. In the case of perfect frequency-channel separation, however, the noise in each channel should be independent, and cross-correlating each channel with every other channel should provide more information about the shape of the channel delay. This information comes at the price of developing a method to correctly combine all this information to form a single channel estimate.

A final detail that is critical to the algorithm's performance is the method used to form a continuous estimate of the channel delay from a sequence of discrete delay estimates. In order to design an equalizing filter based on the delay estimates, it is necessary to be able to specify desired delay characteristics for locations in frequency that are unrelated to the points where the delay shape was estimated. In the algorithm described here, two separate exponential curves were fitted to the estimated delays, one for each of the upper and lower halves of the frequency band. These curves were used to calculate the desired delays at those frequencies required by the function that creates the equalizing filter. For the tests documented in this paper, the distorting channel filter had a "half-C" shape with a quadratically shaped delay curve applied to the lower half band and no curve applied to the upper half band. For this channel shape, the equalizing filters derived from the fitted exponential curves produced were adequate. However, if the expected distorting channel has a more complex shape than a "C" or "half-C" shape such as tested here, then another method of estimating the delay curve at frequency locations other than the tested ones will be necessary.

B. MORE TESTING

The second area of research that is necessary before the channel-correlation algorithm can be considered to be useful for actual situations is to test the algorithm more completely under realistic conditions. Due to time and data constraints, very few tests were done and the only variable that differed among the three test sets was the amount of added noise. Both the input data and the features of the distorting channel need to be varied and the algorithm tested on as wide a variety of data as possible. The results of these tests will be an important guide for the algorithmic development of this equalization method.

The data used in the tests documented here was 500 bursts of Gaussian noise each 100 samples long and separated from adjacent bursts by 100 samples of silence. Real signals would have more structure than Gaussian noise and it is important to

understand how that structure affects the equalization algorithm. Also, real signals would be of varying lengths and have differing percentages of time spent in signal versus silence conditions. It is important that the algorithm be shown to work on a variety of input signals.

Although three different levels of added noise were tested, only one distorting channel was used for all tests. This channel was “half-C” shaped in delay, low-pass shaped in magnitude, and was constant across time. The added noise, if present, was white Gaussian with constant mean and variance. Actual transmission channels come in a wide variety of shapes, with differing noise conditions and time-varying characteristics. The algorithm should be tested under all these differing conditions so that it is well understood in what situations this algorithm is useful.

C. IMPROVED MEASURES OF EFFECTIVENESS

Finally, more research on independent methods to quantify the effectiveness of the equalizing filter needs to be done. An effectiveness measure that does not require additional knowledge of the transmitted signal would be very useful as a check or secondary test on the recovered equalizing filter. Of the measures described in this paper, however, the only one fitting that description, the entropy measure, failed when noise was added to the received signal. The most robust effectiveness measure was the percentage of signal energy outside of signal bursts, but this measure requires knowledge of the location of signal bursts. Perhaps in certain situations, the length of the signal bursts would be known, and their approximate location in the received data could be estimated. In such situations, this measure might be useful. Another area to investigate is if the entropy measure could be improved, possibly by using a less-crude estimate of the probability density function of the samples.

LIST OF REFERENCES

- [1] J. Proakis. *Digital Communications*. McGraw-Hill, New York, 1989.
- [2] D. G. Messerschmitt and E. A. Lee. *Digital Communication*. Kluwer Academic Publishers, Boston, 1989.
- [3] S. Qureshi. Adaptive equalization. *IEEE Communications Magazine*, pages 9–16, March 1992.
- [4] Ellen Kayata Wesel. *Wireless Multimedia Communications*. Addison-Wesley, Reading, MA, 1998.
- [5] H. Vincent Poor and Gregory W. Wornell. *Wireless Communications*. Prentice-Hall Inc, Upper Saddle River, NJ, 1998.
- [6] Lawrence R. Rabiner and Bernard Gold. *Theory and Application of Digital Signal Processing*. Prentice-Hall Inc, Englewood Cliffs, NJ, 1975.
- [7] John G. Proakis and Dimitris G. Manolakis. *Digital Signal Processing*. Prentice-Hall Inc, Upper Saddle River, NJ, 1996.
- [8] Charles W. Therrien. *Discrete Random Signals and Statistical Signal Processing*. Prentice-Hall Inc, Upper Saddle River, NJ, 1992.
- [9] Alan V. Oppenheim and Ronald W. Schafer. *Digital Signal Processing*. Prentice-Hall Inc, Englewood Cliffs, NJ, 1975.
- [10] B. S. Everitt. *The Cambridge Dictionary of Statistics*. Cambridge University Press, 1998.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
8725 John J. Kingman Road., Ste 0944
Ft. Belvoir, VA 22060-6218
2. Dudley Knox Library
Naval Postgraduate School
411 Dyer Rd.
Monterey, CA 93943-5101